Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2011

Eduardo de Oliveira Estanqueiro Rocha

Metodologias para Caracterização de Tráfego em Redes de Comunicações

Methodologies for Traffic Profiling in Communication Networks

Eduardo de Oliveira Estanqueiro Rocha

Metodologias para Caracterização de Tráfego em Redes de Comunicações

Methodologies for Traffic Profiling in Communication Networks

Dissertação apresentada às Universidades de Minho, Aveiro e Porto para cumprimento dos requisitos necessários à obtenção do grau de Doutor no âmbito do doutoramento conjunto MAP-Tele, realizada sob a orientação científica do Doutor Paulo Jorge Salvador Serra Ferreira, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor António Manuel Duarte Nogueira, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Apoio financeiro da Fundação para a Ciência e a Tecnologia através da bolsa SFRH/BD/33256/2007.

o júri / the jury

presidente / president

Prof. Doutor Paulo Jorge de Melo Matias Faria de Vila Real
Professor Catedrático da Universidade de Aveiro (por delegação do Reitor da Universidade de Aveiro)

vogais / examiners committee

Prof. Doutor Rui Jorge Morais Tomaz Valadas
Professor Catedrático do Instituto Superior Técnico da Universidade Técnica de Lisboa

Prof. Doutor Joel José Puga Coelho Rodrigues
Professor Auxiliar da Universidade da Beira Interior

Prof. Doutor Rui Luis Andrade Aguiar
Professor Associado com Agregação da Universidade de Aveiro

Prof. Doutor Paulo Jorge Salvador Serra Ferreira
Professor Auxiliar da Universidade de Aveiro

Prof. Doutor António Manuel Duarte Nogueira
Professor Auxiliar da Universidade de Aveiro

agradecimentos / acknowledgements

I would like to begin to express my deep gratitude to my supervisors for their brilliant guidance, inspiration and scientific excellency. Their continuous challenges, support and friendship provided the perfect environment for carrying out this work.

I cannot also express by words my gratitude for my parents and brother for always being a guiding light in my life. Their examples, support, advices and love are the most precious gift of my life.

To all my friends and colleagues at Farol for three years of great companionship. Special mentions have to be addressed to Vitor Jesus and to Luís Barreto for being always a great source of advices and friendship. To all my MAP-Tele colleagues that accompanied me through these years. To everyone at my research group ATNoG and to IT for great conditions where I could carry out this work. In addition, special thanks to Carlos Ferreira, Carlos Miranda, Ivo Petiz and Fernando Vieira. I would like also to extend my acknowledgment to FCT for the financial support through the grant SFRH/BD/33256/2007.

To Sensei Vitor Gomes for three great years of physical and mental training, for all the challenges he presented me and for always pushing me forward. To everyone at EliteKC for three great years of training and friendship.

Many friends had also a very important role throughout these years but special thanks have to be addressed to Pedro Braumann, Tiago Fonseca, Elsa Rodrigues, Joana Santos, Aneesh Chauhan, Inês Castro and Sérgio and Maria Gonçalves for their incredible support, priceless friendship and for being a part of my family.

Finally, to the woman of my life Cornelia Rinn for being such a great companion and support though these years. For never letting me go down and for understanding me as no-one ever did. To my son for being already a light in my life. This thesis is dedicated to you and your mother.

palavras-chave

Tráfego Internet, Aplicações Internet, Ataques na Internet, Perfis de Tráfego, Análise Multi-Escalar.

resumo

A Internet pode ser vista como uma plataforma em constante evolução onde novos e diferentes serviços e aplicações estão constantemente a emergir. De facto, muitas das aplicações dominantes, tais como redes sociais, apareceram recentemente tendo sido rapidamente adotadas pela comunidade de utilizadores da Internet. Todas estas novas aplicações requerem a implementação de novos protocolos que apresentam diferentes requisitos de rede de acordo com o serviço que implementam. Toda esta diversidade levou à necessidade de construção eficiente de perfis de utilizadores através do mapeamento do tráfego destes na aplicação que o originou. Várias tarefas de gestão de redes tais como a otimização de recursos, desempenho da rede, personalização de serviços e segurança podem beneficiar de um eficiente mapeamento de tráfego. No entanto, esta é uma tarefa difícil devido à complexidade inerente dos protocolos existentes e a várias restrições que impedem a análise dos conteúdos do tráfego. De facto, muitas tecnologias, tais como a encriptação do tráfego, são amplamente utilizadas para proteger a confidencialidade e integridade das comunicações na Internet. Por outro lado, várias limitações legais impedem também a análise dos tráfego de utilizadores da Internet de modo a proteger a sua confidencialidade e privacidade. Consequentemente, novas metodologias de discriminação de tráfego são necessárias para uma eficiente construção de perfis de tráfego e de utilizadores. Esta tese propõe várias metodologias que permitem uma construção precise de perfis de tráfego e que operam eficientemente sob as várias restrições que foram atrás mencionadas. Através da análise das componentes de frequência presentes no tráfego capturado e da avaliação da presença dos vários eventos causados e relacionados com os utilizadores e com a própria rede, as metodologias propostas são capazes de construir um perfil para cada uma das aplicações da Internet estudadas. O uso de vários modelos probabilísticos permite uma associação exata do tráfego analisado à aplicação correspondente. Várias extensões podem ser feitas às metodologias propostas de modo a permitir a identificação de perfis ilícitos escondidos em comunicações legítimas bem como a classificação de tráfego em tempo real. Um novo paradigma para a gestão de redes com e sem fios é também proposto, em que através da análise de métricas da camada 2 e das várias componentes de frequência presentes é possível uma construção eficiente de perfis dos utilizadores ligados em termos das aplicações-web usadas. Por fim, alguns cenários de utilização vão ser apresentados e discutidos.

keywords Internet Traffic, Internet Applications, Internet Attacks, Traffic Profiling, Multi-Scale Analysis

abstract Nowadays, the Internet can be seen as an ever-changing platform where new and different types of services and applications are constantly emerging. In fact, many of the existing dominant applications, such as social networks, have appeared recently, being rapidly adopted by the user community. All these new applications required the implementation of novel communication protocols that present different network requirements, according to the service they deploy. All this diversity and novelty has lead to an increasing need of accurately profiling Internet users, by mapping their traffic to the originating application, in order to improve many network management tasks such as resources optimization, network performance, service personalization and security. However, accurately mapping traffic to its originating application is a difficult task due to the inherent complexity of existing network protocols and to several restrictions that prevent the analysis of the contents of the generated traffic. In fact, many technologies, such as traffic encryption, are widely deployed to assure and protect the confidentiality and integrity of communications over the Internet. On the other hand, many legal constraints also forbid the analysis of the clients' traffic in order to protect their confidentiality and privacy. Consequently, novel traffic discrimination methodologies are necessary for an accurate traffic classification and user profiling. This thesis proposes several identification methodologies for an accurate Internet traffic profiling while coping with the different mentioned restrictions and with the existing encryption techniques. By analyzing the several frequency components present in the captured traffic and inferring the presence of the different network and user related events, the proposed approaches are able to create a profile for each one of the analyzed Internet applications. The use of several probabilistic models will allow the accurate association of the analyzed traffic to the corresponding application. Several enhancements will also be proposed in order to allow the identification of hidden illicit patterns and the real-time classification of captured traffic. In addition, a new network management paradigm for wired and wireless networks will be proposed. The analysis of the layer 2 traffic metrics and the different frequency components that are present in the captured traffic allows an efficient user profiling in terms of the used web-application. Finally, some usage scenarios for these methodologies will be presented and discussed.

Contents

List of Figures

List of Tables

Acronyms

AP Access Point

CCS Conversation Content Sequence

CDF Cumulative Distribution Function

CI Confidence Interval

CoS Class-of-Service

CWT Continuous Wavelet Transform

DoS Denial-of-Service

DDoS Distributed Denial-of-Service

DWT Discrete Wavelet Transform

DBSCAN Density Based Spatial Clustering of Applications with Noise

EWMA Exponentially Weighted Moving Average

FCA Formal Concept Analysis

FTP File Transfer Protocol

HTTP Hypertext Transfer Protocol

IAT Inter-Arrival Time

IDSes Intrusion Detection Systems

IMAP Internet Message Access Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

IPTV Internet Protocol Television

IRC Internet Relay Chat

k-NN k-Nearest Neighbors

ML Machine Learning

MSE Mean Squared Error

PCA Principal Component Analysis

PDF Probability Distribution Function

P2P Peer-to-Peer

POP Post Office Protocol

QoS Quality-of-Service

SFS Sequential Forward Selection

SPs Service Providers

SMTP Simple Mail Transfer Protocol

SSH Secure Shell

TCP Transmission Control Protocol

UA University of Aveiro

UDP User Datagram Protocol

WB Web-Browsing

WiFi Wireless Fidelity

Chapter 1

Introduction

The Internet can be seen as a constantly evolving domain that became, in recent years, the most powerful communications platform on the Planet, being an excellent means for accessing and sharing different types of information, services and applications. In fact, nowadays, Internet users are able to watch on-line videos, watch Internet Television (IPTV) broadcasts, use chatting applications, make voice and video calls and many more. The Network also began to be intensively used for a wide range of military, research, commercial and financial purposes: nowadays, it is possible to perform several financial transactions either using on-line banking services, which are widely available, or accessing services provided by companies adopting pure e-commerce based business models. These companies have been able to obtain increasing revenues in the last years, proving that such business models are becoming widely adopted by the consumers. In addition, the recent emergence of Web 2.0 services changed the Internet itself and the way users interact with it. Indeed, the contents available on the Internet are more user-centered, since users are now active producers of contents and information, being also able to share such contents with the on-line community. The emergence of a vast set of applications based on this new Internet concept, such as social networks, have contributed to this revolution and such applications are nowadays dominant. Other technologies, such as HTML 5 and IPv6, also promise a complete revolution on the way users interact with the Internet and experience its contents. Furthermore, the increasing relevance of cloud computing and of the applications based on this novel paradigm present a significant challenge to the Internet and its infrastructures by requiring significant storage and connectivity demands. As a final note and, despite the promising features and revolutions that these applications present, several security issues and vulnerabilities emerge [GWS11]. One can then conclude that the Internet is currently facing an unprecedented challenge [BDF+09].

As the Internet grew in size and complexity, the challenge of provisioning, managing and securing it became intrinsically linked to the understanding of Internet applications and of the generated traffic. On the other hand, all these new applications and services implemented novel communication paradigms that require distinct resources from the network, according

to the application or service they implement. Currently, satisfying the clients' needs is absolutely mandatory and the ability of mapping all this traffic to the source application offers invaluable informations about which resources to allocate to the traffic, which is essential to guarantee a high quality and availability of the network services. Consequently, in recent years, building accurate user and traffic profiles became tasks of critical importance for many different networking aspects, such as network performance, network resources management, service personalization and security. For instance, Service Providers (SPs) and network managers can more easily infer the bandwidth and delay requirements, assigning the corresponding traffic to the most adequate class of service and, thus, offering a more satisfying Quality-of-Service (QoS). Network performance can also be enhanced by inferring resources requirements from the usage profile of each client, thus allowing a better management of the network resources. In addition, by predicting future user requirements, network resources can also be timely allocated in order to prevent their future saturation. Service personalization and content customization can also be greatly improved since the delivery of related contents and applications is eased by inferring the applications and contents that are more requested by network users. Network security can also be improved: by accurately mapping traffic to its originating application, flows generated by illicit applications or flows presenting suspicious patterns can be more easily detected. Therefore, by performing a timely detection of security attacks or compromised hosts, it is easier to achieve a better protection of the remaining connected clients and critical network infrastructures, thus avoiding incommensurable monetary losses [Ins11]. Many methodologies have been proposed to address the traffic classification issue, but they had to evolve together with the complexity of the emerging Internet applications and services and of the Internet itself [DP10]. Most of the existing methodologies are based on the statistical analysis of Internet traffic or on the deep-inspection of the contents of the packets. However, many considerable constraints, such as traffic encryption and legal restrictions, prevent the efficient deployment of these classification approaches. The main reason behind these constraints is the protection of the confidentiality and privacy of users' on-line communications. In fact, privacy is a key aspect when considering traffic analysis and many approaches have been proposed to cope with imposed restrictions [MM10].

Indeed, in the last years, we have also assisted to a dramatic increase on the number and variety of Internet-based attacks [Sec11a]. Such increase, illustrated in figure 1.1, was caused by the growing interest of the hacker community, who shifted their focus from exhibitionism purposes to financial gains due to the wide adoption of e-commerce business models and e-banking services. The aim is now to profit from existing vulnerabilities on the mentioned services, causing considerable financial losses to both clients and companies. As a consequence, users' confidence in these services and their providing companies can be seriously damaged, with consequent incommensurable losses. As reported in [Ins11], in 2010 the organizational cost of a data breach rose to $7.2 million. In addition, the evolution of the attack kits used

by perpetrators made them more professional, thus more marketable and easy to deploy. The attacks carried out by such attack toolkits have also become more stealth, more distributed and more difficult to detect, prevent and mitigate. Such evolution is illustrated in figure 1.2, which shows the evolution and emergence of novel attack toolkits as well as their growing sophistication.

Botnets emerged as the cornerstone of on-line criminal activities by allowing the use of several compromised hosts, under the control of a single master entity - the *bot-master*, for specific illicit purposes. These include launching Distributed Denial-of-Service attacks (DDos), sending spam and phishing e-mails, stealing private information, among others. Due to their volume, diverse capabilities and robustness, they pose a significant and growing threat to enterprise networks, costumers and to the Internet as a whole [Ell10]. Detecting *botnet* compromised machines, the so called *bots*, is a difficult task and traditional network and host security systems, such as *firewalls* and Intrusion Detection Systems (IDSes), are unable to successfully complete it due to the stealth and distributed nature of *botnets*. Moreover, the inability of these systems to operate in encrypted traffic scenarios also prevents them from performing an accurate and timely detection of the traffic generated by the compromised hosts. Indeed, the communications between compromised hosts and their *masters* use traffic encryption for hindering purposes. Moreover, such communications can run on top of well-known protocols, such as the ubiquitous HTTP protocol, thus preventing security systems from detecting such traffic.

It can be concluded that new paradigms and methodologies for the analysis of Internet traffic and for the detection of illicit traffic are necessary that can cope with all the mentioned restrictions are necessary. Our thesis addresses this issue and several classification methodologies are proposed. Such approaches analyze different traffic metrics extracted from the captured traffic and explore the different frequency components present in order to obtain Multi-Scale Application Signatures that enable the accurate association of unknown traffic with the corresponding Internet applications. In addition, such approaches also allow an accurate identification of low-impact and stealth anomalies.

1.1 Motivation

The issue of traffic classification has been recently object of several research works. The need for novel classification methodologies that can cope with the complexity of existing networks, with their increasing capacity and bandwidth and with the emergence of several novel Internet applications was one of the motivations behind our work. In addition, the need of accurately and timely identifying illicit traffic or traffic presenting suspicious patterns was also an important motivation for our work. Besides, there is a lack of methodologies that characterize Internet traffic in terms of the generated traffic patterns, encompassing

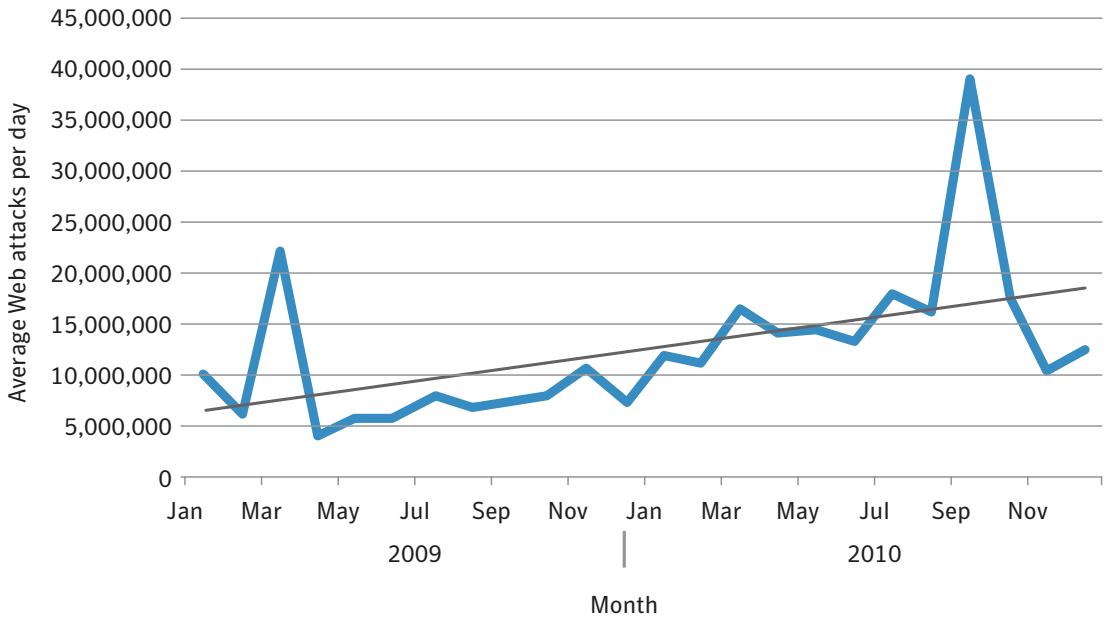

Figure 1.1: Average Web-based attacks per day, by month, 2009–2010 (source [Sec11a]).

different traffic metrics, together with a lack of approaches decomposing network traffic into the several frequency components and evaluating the frequency components and dynamics of the traffic generated by the several Internet applications. It is known that traffic of any Internet application is generated and shaped by several events and mechanisms occurring in different components of the frequency spectrum. Such components include low frequency events, such as the user interactions or requests, mid-range frequency events, which include the traffic sessions, created by the user requests, and high frequency events, which encompass events such as packet arrivals. Since different Internet applications require different user interactions, creating different frequency events and components, obviously different traffic patterns are generated, allowing us to create an unique frequency spectrum profile that can be seen as a signature for each Internet application. Existing classification methodologies either need to inspect the contents of captured traffic, thus not coping with the several privacy restrictions imposed by Service Providers (SPs) and with the encryption of such contents which prevent their analysis, or need to analyze the complete flow in order to perform some form of statistical analysis over the captured traffic.

In addition, we also propose a new network management paradigm based on the accurate profiling of Internet users. The proposed paradigm evaluates the frequency components and the dynamics of the captured Internet traffic by performing a multi-scale decomposition. This consists of analyzing traffic in several scales, *i.e* different aggregation levels, in order to capture the above mentioned mechanisms. Several differentiating regions can be defined in whole

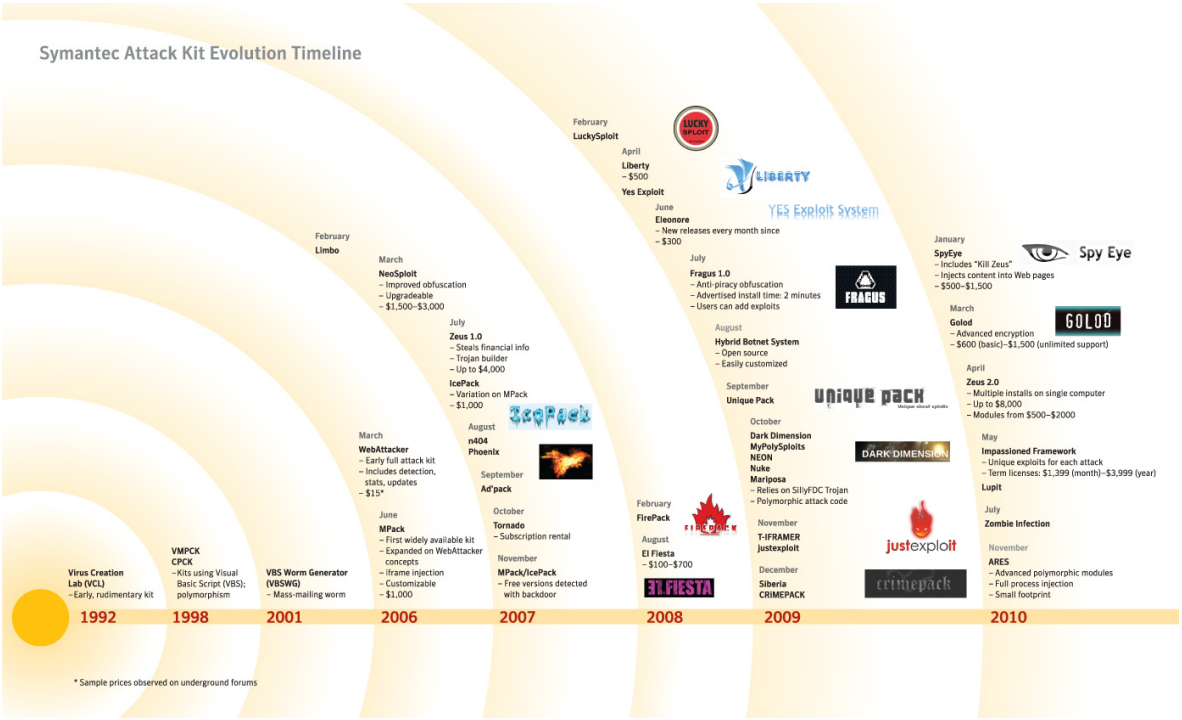

Figure 1.2: Evolution of attack toolkits and notable innovations (source [Sec11b]).

frequency spectrum and the analyzed traffic can be accurately mapped to the corresponding Internet application. The obtained results allow us to conclude that the proposed approach enables an accurate traffic mapping and user profiling. As the approach is based on the analysis of different traffic metrics that can be extracted according to each classification scenario, it complies with the several existing traffic analysis restrictions and can be deployed in encrypted traffic scenarios, where the contents of the packets are not available for analysis, or scenarios where strong restrictions prevent the analysis of the network traffic.

1.2 Objectives

There are several objectives for this PhD work. The first one is the implementation of classification methodologies that allow an accurate traffic classification and an accurate identification of Internet-based attacks. Such tasks are essential for building user profiles, which is a critical task in many crucial network management tasks. The proposed methodologies must also cope with the different limitations imposed by many existing privacy and legal restrictions that prevent the inspection of the payloads of the captured traffic. In addition, proposed approaches must cope with traffic encryption, since it is a widely deployed technique for protecting the confidentiality of on-line communications by encrypting the packets contents.

The implementation of user profiling techniques in wired and wireless network scenarios is another objective of this work. By performing a promiscuous monitoring of all connected hosts, different traffic metrics will be measured/inferred, allowing the definition of a User Profile as the set of Internet applications that are used by a specific user.

The identification of profile changes and of hidden illicit patterns constitute an important objective in order to detect stealth and low-impact attacks. Finally, the timely association of captured traffic to the generating Internet application and the timely identification of Internet-based attacks are also crucial objectives of this PhD. To address this issue, different non-sampled traffic metrics must be inferred in order to reduce the amount of time required for obtaining an accurate traffic classification.

1.3 Applicability Scenarios

In this section, we propose and describe a network management and profiling platform where the different approaches proposed in this thesis can be deployed. In addition, some possible deployment scenarios will also be presented.

An overall diagram with the different components of the envisioned platform is presented in figure 1.3. To begin with, several network probes can be deployed to monitor each network segment in order to capture the sent and received traffic. Measurements can also be made at other critical network points, such as ingress and egress nodes, where all traffic has to flow through. Several traffic metrics can then be extracted from the captured traffic, being subsequently decomposed in their frequency/time components. From such decomposition, a wavelet spectrum is obtained and a profile, which depicts several frequency components, is built for the traffic generated in each host by each Internet application. When analyzing *known* traffic, *i.e* traffic whose generating application is known, a *Multi-Scale Signature* is obtained for each application class. These signatures allow the association of unknown traffic to the generating application and can be stored in a database for future traffic classification. A classifier is used to associate captured and unknown traffic with the corresponding application class and after a validation process, which can be performed by network managers, the corresponding (and already stored) *Multi-Scale Application Signature* can be updated, which enables an adaption to changing network conditions, such as bandwidth, and consequently to changing applications profiles. The classification is then passed to the user profiling module that creates and updates the profile of each user accordingly. Finally, the network management module interacts with classifier and user profiling modules in order to perform (i) counter-measures when illicit traffic is detected or (ii) other optimization tasks on the monitored network in order to improve its performance.

The Classifier module will be addressed in chapters 4 to 6, which propose different methodologies to perform an accurate classification of the analyzed traffic to the corresponding Inter-

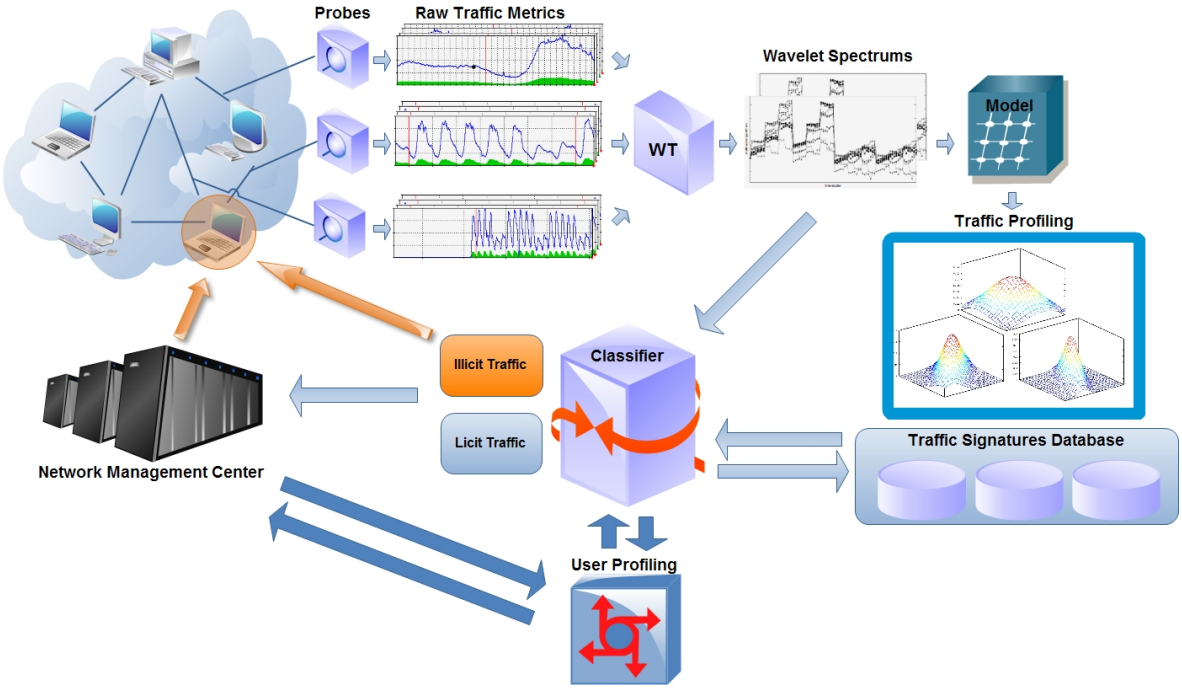

Figure 1.3: Proposed architecture.

net application. Such methodologies cope with the several existing privacy, legal and technical restrictions, being also able to analyze encrypted traffic and identifying the generating Internet application. These features enable the proposed platform to be deployed in scenarios having stringent restrictions.

The User Profiling Module will be addressed in chapter 7, which proposes an approach for the accurate profiling of the users that are connected to wired and wireless networks. User Profiles are, in our work, defined as the set of used web-applications and, by performing an analysis of the several frequency components present in the traffic generated by a local host, the proposed approach can create accurate user profiles.

Several usage scenarios can be envisioned for the proposed profiling platform. To begin with, our platform can be used for profiling users connected in wired and wireless networks. Indeed, the proposed approach is able to analyze traffic sent over secure wireless networks by deploying probes that do not register in the network(s) and can promiscuously monitor the different connected clients. The monitoring and profiling platform can be also deployed for the identification of compromised hosts inside a network. The identification of such hosts has a critical importance, since traditional network defense mechanisms and intrusion protection systems only look for attacks coming from outside the network. Therefore, compromised hosts inside the network are free to run all kinds of illicit activities and need to be identified in order to stop them. This can be achieved by monitoring each host and issuing an alert whenever

a pattern of illicit or suspicious traffic is detected at any scale of analysis: in this case, the proposed framework univocally identifies the compromised host. Several counter-measures, which include isolating that host from the network or simply shutting it down, can then be taken. Such use case is illustrated in figure 1.3.

As already mentioned, our traffic classification approaches are immune to existing privacy restrictions and, consequently, the proposed platform can be deployed in encrypted traffic scenarios where the payload of the packets is not accessible. Consequently, this paradigm is also suitable for the analysis of Virtual Private Network (VPN) connections, of Internet tunnels and of secured wireless networks.

1.4 Structure of the Thesis

The thesis is divided in eight chapters.

In chapter 2, the most relevant related work is presented. The chapter covers several research topics that are related to this thesis, including traffic classification, intrusion and attacks detection and the detection of traffic generated by *botnets* and by compromised hosts. To conclude this chapter, some important related to User Profiling is discussed.

Chapter 3 presents some relevant definitions that are common to all presented traffic models and classification approaches. These include the definition of *data-stream* as a traffic grouping methodology, representing the object of analysis, modeling and classification. A presentation of the legitimate Internet applications that will be studied, as well as of the illicit applications that will be emulated, is also provided in this chapter. The different generated and captured traffic patterns are described and the capturing methodology is also explained. Subsequently, a comparison between some of the most used signal frequency analysis methodologies, together with an explanation of the advantages and drawbacks associated to each decomposition approach, is also given. Continuous and discrete transforms are presented and discussed, highlighting the advantages and issues associated to each transform. Our multi-scale analysis approach is then presented, explaining how such analysis is enabled by Continuous or Discrete Wavelet Transforms (CWTs or DWTs). Subsequently, some realted preliminary definitions are also given, since they will be intensively used in subsequent chapters.

Chapter 4 presents a classification methodology for grouping the traffic presenting similar behaviors over the analyzed range of decomposition scales. The proposed approach uses unsupervised clustering algorithms for analyzing the high frequency components of the several studied Internet applications and grouping the ones with similar behaviors. The first and second order moment of analysis are used and the obtained results show that unknown traffic can be accurately assigned to the corresponding application. In addition, an accurate identification of stealth and low-impact Internet attacks was achieved. The accuracy of the

approach was evaluated using different classification metrics.

Several proposed probabilistic modeling approaches are presented and defined in chapter 5. We start by presenting unidimensional probabilistic models that include Gaussian and generic probability distributions. Two different approaches were used to select the decomposition scales that are used for traffic classification. The chapter starts by presenting an approach that uses the first decomposition scales for modeling the high frequency components generated by the different Internet applications and by some relevant, low-impact and stealth anomalies. After a short introduction, an algorithm for selecting the most appropriate decomposition scales is presented. For each decomposition scale, the algorithm uses known traffic for inferring the parameters of the distributions of each Internet application and chooses the ones in which the distributions are more separated. The unidimensional probabilistic models used for associating unknown traffic to the corresponding Internet application are then presented. Such models analyze each scale of analysis separately, inferring distributions for each Internet application in the different decomposition scales and then computing a final value. Subsequently, multidimensional approaches that map each one of the selected decomposition scales into a dimension in order to generate a n-dimensional space are presented. These approaches enable the analysis of the correlations between the used decomposition scales, in order to infer more accurate distributions. The models that are used are multidimensional generic and Gaussian approaches. Finally, a discussion of the obtained results and a comparison between the different approaches are provided.

Chapter 6 presents an enhancement that can be made to all proposed classification approaches in order to increase the classification accuracy. Such enhancements enable the identification of profile changes in the traffic generated by an Internet application, allowing the identification of hidden illicit patters, or the real-time traffic classification of network traffic. The results obtained show that the proposed enhancements can be used for accurately identifying hidden illicit patterns and for enabling a timely traffic classification and a timely identification of low-impact and stealth anomalies.

Chapter 7 presents a novel user profiling paradigm for wired and wireless networks based on the analysis of different traffic metrics, according to the restriction of the classification scenario. In the proposed scenario, promiscuous monitoring probes that do not authenticate with the Access Point of the monitored network are used for capturing the traffic each host sends and receives. Since the monitoring probes do not register in the monitored network, they cannot be detected by the monitored host neither by the Access Point. Layer 2 traffic metrics are captured from the traffic generated and received by each one of the hosts of the monitored network and, by using Continuous Wavelet Transforms (CWTs), appropriate scalograms are constructed in order to depict the most important frequency components of the captured traffic. Subsequently, the creation of differentiating regions associated to each one of the studied web-applications (indicating the characteristic frequency components of

the traffic it generates) allows an accurate identification of unknown traffic. In this way, it is possible to characterize the connected hosts in terms of the web-applications they use, which constitutes our definition of user profile. The obtained results prove that the proposed approach is suitable for creating such usage profiles.

Finally, Chapter 8 presents the most relevant conclusions as well as some directions for future work.

1.5 Major Contributions

The following constitute the major contributions achieved with this thesis:

- Proposal and analysis of a traffic classification approach based on the clustering of the multi-scale decomposition estimators. This approach models the first and second order multi-scale components and is able of accurately grouping the traffic presenting the similar frequency components. In this manner, an efficient grouping of the traffic presenting the same multi-scale behavior is achieved and an accurate traffic discrimination is obtained together with the identification of some of the most used Internet attacks. This approach was presented in [RSN09a], [RSN09b], [RSN11e];

- Proposal and analysis of a legitimate and illicit traffic discrimination methodology based on unidimensional probabilistic models. Two different approaches were used for selecting the most appropriate decomposition scales. The first consisted in analyzing the first scales in an attempt to model the high-frequency events present in the captured traffic and is presented in [RSN10]. The second approach consisted in deploying an algorithm for choosing the decomposition scales where the distributions are more separated. In this manner, we can optimize our methodologies. Two different models were used for assigning unknown Internet traffic to the corresponding application. These assume that the distributions generated by the multi-scale decomposition estimators can be modeled with Gaussian approaches, while the second uses generic probabilistic approaches for classifying captured traffic. Once again, an accurate traffic mapping was achieved together with an accurate identification of two widely deployed Internet attacks. The proposed models were presented in [RSNR11];

- Proposal of a legitimate and illicit traffic identification model based on the use of multidimensional probabilistic models for assigning Internet traffic to the generating application. The previously mentioned algorithm was used for choosing the most suitable decomposition scales for an accurate traffic classification and two multidimensional models were used. The first one uses multivariate Gaussian distributions for modeling the multi-scale traffic components and was presented in [RSN11b], while the second uses multidimensional generic approaches and was presented in [RSN11c];

- Proposal of an enhancement for the classification methodologies which consists on the usage of several sliding classification windows that allow the identification of hidden illicit patterns embedded inside legitimate communications and was presented on [RSN11c]. The use of non-sampled traffic metrics allowed us to achieve a timely traffic classification together with an accurate identification of some significant illicit traffic. The mentioned work is presented in [RSN11d];

- Proposal and analysis of a user profiling approach for wired and wireless networks. The approach uses promiscuous monitoring probes that capture the traffic generated by each connected client in every network segment. Several traffic metrics are then extracted and decomposed and the obtained parameters can be accurately assigned to the generating Internet application. The proposed approach can also be deployed for monitoring different hosts connected in different wireless networks since the monitoring probes do not authenticate with the Access Point(s) of the monitored network(s). Layer 2 traffic metrics are then extracted and decomposed using a CWT and the analysis of the resulting scalograms allows an evaluation of the different frequency components, enabling the accurate assignment of the analyzed traffic to its corresponding application. The profiling approach was presented in [RSN11a].

Chapter 2

State-of-the-art

2.1 Introduction

This chapter presents the most relevant work that has been carried out in the fields addressed by this PhD. Several areas are related to the classification methodologies that we will propose. So, let us start by presenting the the most relevant work and the different methodologies that have been proposed in the traffic classification area. After that, the field of intrusion and security attacks detection is discussed, followed by a discussion of the most relevant work on the field of *Botnets* detection. Finally, the need of an accurate user profiling is discussed, together with the presentation of the most relevant related work.

2.2 Traffic Classification and Network Anomaly Detection

The problem of classifying Internet traffic has been studied for many years and constitutes a very active research field. In fact, the ability to accurately associate captured traffic with its source application is of critical importance for many network activities. For instance, network administrators can easily build and identify application usage trends that can be essential for many network management tasks, including traffic engineering, network links optimization and service personalization [CKS+09]. The identification of emerging applications can also be achieved by an accurate traffic mapping, which can help network administrators in identifying applications that can change the network resources demands and, consequently, lead to a saturation of some of those resources. Network security can be also improved, because an accurate mapping between traffic and its source application will certainly lead to an easier identification of illicit traffic or traffic presenting an anomalous behavior.

The main concept behind traffic classification is the ability to infer, from captured traffic, the necessary characteristics that will allow an accurate association of the traffic to its originating Internet application. Many approaches have been proposed over the years, which had also to evolve together with the increasing complexity of existing and emerging Internet

applications and services. In the following sub-sections, the different methodologies will be presented, together with a discussion of their main advantages and drawbacks and the most relevant work that has been done so far.

2.2.1 Port-Based Classification Approaches

Ports numbers are divided in three ranges: Well Known Ports, Registered Ports and Dynamic and/or Private Ports. The Well-Known range spans from 0 to 1023 and corresponds to ports reserved for privileged use, while Registered Ports are those from 1024 to 49151. Finally, Dynamic and/or Private Ports are the ones located between 49152 and 65535, *inclusive*. These numbers are assigned by IANA [IAN11].

The first deployed classification approach explores the concept that each Internet application uses a specific port number. For instance, HTTP traffic uses port number 80, while DNS uses port number 53; SMTP runs on top of port 25 and FTP uses ports 21 and 22. Therefore, a simple association between the used port and the corresponding services can be performed in order to classify traffic. Some works using this approach were proposed, being able to achieve accurate results [MKK+01].

However, in the last years this approach became inefficient since many emerging applications started to use ephemeral ports in an attempt to disguise themselves by using ports that are usually associated to different protocols. Peer-to-Peer protocols and voice or video transmission protocols started to exhibit such behaviors in order to bypass proxies and firewalls. A study conducted by Madhukar et. al. [MW06] proved that port-based analysis no longer provided accurate results when compared to other identification methods, since unknown traffic varied between 40% to 65% of the total traffic. This study also confirmed that unknown traffic was more evident at night periods, which might suggest that this was generated by Peer-to-Peer (P2P) applications. In a related work, Sen et. al. [SSW04] stated that the default port of the Kazaa protocol only accounted for 30% of the total traffic generated by this protocol. In [KBB+03], the authors developed several classification heuristics that allowed them to identify P2P traffic running over nonstandard ports. The authors concluded that, according to the protocol and metric that was used, approximately 30% to 70% of P2P traffic could not be identified using standard ports. In [KBB+04] the authors went beyond the known port classification limitation for identifying P2P traffic and developed a framework and heuristics to measure camouflaged P2P traffic. Their work consisted in reverse engineering of the protocols and identification of characteristic strings in the payload. The results obtained showed that P2P applications evolved to use arbitrary ports for communication. Finally, in [MP05] an analysis and quantification of the errors due to this classification approach were presented and the results obtained showed that more than 28% of the captured traffic could not be classified.

2.2.2 Payload-Based Classification Approaches

One of the most accurate classification approaches is based on the fact that many Internet protocols and applications use characteristic strings in the payloads of the generated packets that can actually distinguish them. Such strings are also known as *digital signatures* and consist of specific byte sequences. Therefore, this approach is based on the inspection of the payloads of the captured packets searching for *digital signatures* that can be used to identify the generating Internet application [MW06].

In one of the first works, presented in [SSW04], the authors proposed application level signatures for an efficient identification of P2P traffic. The authors analyzed the available documentation and packet-level traces from the different existing P2P clients in order to obtain the application-level signatures, which were then used to develop on-line filters that could efficiently track P2P traffic in high-speed network links. Authors were able to achieve very accurate classification results, with less than 5% of false positives and false negatives. However, the approach required a previous knowledge of each application in order to develop the corresponding signatures, which prevents this approach from automatically adapting itself to new/emergent applications.

A very important work [KBB+04] addressed the reports that claimed a significant decrease in P2P file-sharing traffic. The authors started by measuring traffic from all known P2P protocols and, using reverse-engineering, analyzed these protocols in order to identify characteristic strings in the payload, like the ones shown in table 2.1. Several classification heuristics were then proposed to accurately determine if the analyzed traffic was generated by a P2P protocol or not. These included analyzing the source or destination port and determining if it matched "known P2P ports", in which case the flow was tagged as P2P. Subsequently, the authors compared the payload of each packet against the obtained characteristic signatures, which allowed them to determine the exact P2P protocol. Their findings contradicted the reports that claimed a decrease on the volume of P2P traffic and also pointed some obstacles for an accurate identification of this type of traffic.

In [MP05], authors used payload analysis to quantify the errors associated to port-based classification approaches. The traffic used for classification was captured from a site referred

Table 2.1: P2P protocols and their characteristic strings used in [KBB+04].

P2P Protocol	String	Transport Protocol
eDonkey2000	0xe3, 0xc5	TCP/UDP
Fasttrack	"GIVE" / 0x270000002980	TCP/UDP
BitTorrent	"0x13Bit"	TCP
Gnutella	"GNUT" / "GIVE" / "GND"	TCP/UDP
MP2P	GO!!, MD5, SIZ0x20	TCP
Direct Connect	"$MyN","$Dir" / "$SR"	TCP/UDP

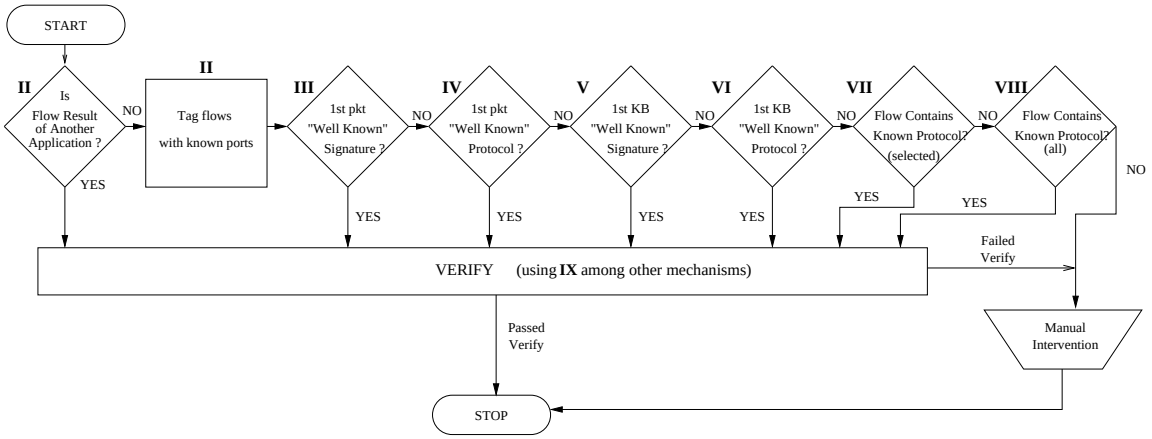

Figure 2.1: Classification approach used in [MP05].

to as *Genome Campus*, hosting several Biology-related facilities. The captured traffic was grouped into flows for a more-efficient processing of the collected information and for retrieving context for an appropriate identification of the network application that generated each flow. Flows were classified according to the approach illustrated in figure 2.1 and very accurate classification results were obtained.

In [HSSW05], authors explored the automatic extraction of application signatures by applying three statistical machine learning algorithms, namely Naïve Bayes, AdaBoost and Regularized Maximum Entropy models. In this manner, their classification approach was able to scale in order to allow traffic identification on high-speed links. The studied applications were FTP control, SMTP, POP3, IMAP, HTTPS, HTTP and SSH, and were chosen because they cover a wide range of application classes and it was very easy to obtain the required pre-classified training set, since these applications still use mainly their default ports. The authors also evaluated the durability of the extracted signatures by classifying traffic traces collected 7 months after their first data set. The classification errors only slightly increased, which indicated that the classifiers maintained a good performance and the signatures could be used for long time periods.

In a study addressing network traffic measurements for P2P applications [MW06], the authors compared three different classification methods: port-based, payload signatures and transport-layer analysis. Traffic traces collected during 2 years at the University of Calgary were used to verify the accuracy of the three different methods. As already mentioned in section 2.2.1, the port-based approach was unable to classify 30% to 70% of the captured traffic. Authors then used several application signatures to identify the P2P protocol that was behind each captured packet. The available documentation and packet-level traces were used to infer the most accurate signatures and this approach was used to establish ground-truth,

Table 2.2: P2P protocols and their characteristic strings used in [MW06].

P2P Protocol	String
Gnutella2	"GNUTELLA"
KaZaA	"X-Kazaa"
BitTorrent	".BitTorrent"

i.e, results that are close to the ones determined by using manual inspection. Their analysis focused on P2P protocols transmitting data over TCP, which were the vast majority, and on reassembled TCP streams, so that signatures that span several packets could be detected. In addition, signature matching needs only to be performed once per stream, which reduces analysis overhead. The analyzed P2P applications were Gnutella2, KaZaA and BitTorrent and the used signatures are shown in table 2.2. The results obtained with these signatures were quite accurate and this method did not classify any non-P2P flow as P2P. The authors also pointed out some disadvantages of using this approach, including the fact that many privacy regulations prevent the deep-inspection of the contents of the captured traffic and the need of knowing in advance which applications are going to be identified. In addition, many P2P protocols started to use encryption, which turned this approach into an infeasible one.

Many issues are raised when deploying such an approach. In fact, many privacy restrictions prevent the analysis of the packets contents, while many technical issues appear when using this classification approach in high-speed network links [MM10]. In addition, the constant updates that are required to keep all the signatures databases accurate and suitable for traffic classification constitute a significant drawback of this approach. Finally, traffic encryption that is widely deployed to assure the confidentiality and integrity of on-line communications by encrypting the traffic contents prevents the use of this approach.

2.2.3 Statistical-Based Classification Approaches

The study of the statistical properties of the traffic flows, which is based on the fact that different applications typically generate different traffic patterns that enable the identification of their underlying protocols, can be a very efficient identification approach, even when unknown P2P protocols are included [MW06, HCL08].

A very important work using this classification paradigm presented a methodology for the identification of P2P traffic at the transport layer based on the connection patterns and was proposed in [KBFc04]. It addressed the ability of P2P applications to disguise their presence by using arbitrary ports and the many restrictions preventing payload-based approaches. In this impressive work, the authors started by pointing out the different limitations associated to payload-based classification approaches and then presented their nonpayload P2P traffic classification methodology, which used two heuristics. The first one examines source-

destination IP pairs that use both TCP and UDP for data transfer, since most of the studied P2P protocols use both layer 4 protocols. In fact, control traffic, such as queries and query-replies, use UDP, while data transfers run on top of TCP. The second heuristic is based on how peers connect to each other and the authors examined all source $\{srcIP, srcPort\}$ and destination $\{destIP, destPort\}$ pairs. The pairs for which the number of connected IPs is equal to the number of connected ports are considered as P2P. This methodology was able to identify more than 90% of P2P bytes, even with bit-rates as high as 220 Mbps.

In a ground-breaking work [RSSD04], authors presented a methodology for associating captured traffic to its Class-of-Service (CoS) and described the requirements and associated challenges, outlining a solution framework for measurement based classification of traffic for QoS purposes. These authors stated that the chosen signatures are insensitive to the used application layer protocol but are able to determine how an application is used: interactively or for bulk-data transport. These signatures can then be used to determine the CoS for each IP packet. The work focused on four broad application classes:

- *Interactive:* this class encompasses traffic which is required by a user to perform multiple real-time interactions with a remote system;
- *Bulk data transfer:* traffic used to transfer large data volumes over the network without any real-time constraints;
- *Streaming*: multimedia traffic with real-time constraints;
- *Transactional*: traffic which is used in a small number of request response pairs which can represent a transaction.

Several features were extracted from captured traffic at different levels: packet, flow and connection levels. Several feature vectors were built and two methods were then used for classification: Nearest Neighbors (NN) and Linear Discriminant Analysis (LDA). Large traffic traces from different network locations were used to assess the accuracy of the methodology, which presented low error rates.

A novel classification paradigm based on the identification of host behavior patterns at the transport layer was proposed in an impressive work [KPF05] with the title BLINC. The mentioned traffic patterns are analyzed at three levels of increasing detail. The first one, the social level, captures the behavior of a host in terms of the number of hosts it communicates with, which the authors refer to as *popularity*. The information required for analyzing such level consists only on the source and destination IP addresses. The functional level captures the functional role of the host in the network, *i.e*, if it is a consumer or a provider of a service or if it participates in collaborative communications. The additional information required for analyzing the functional role is the source port, because if a single port is used for most of the communications, it is likely that the host is providing a service offered in that port.

Finally, the application level captures the transport layer interactions between hosts in order to identify the source application. The authors use the 4-tuple (IP addresses and ports) and include additional flow information, such as the number of packets or bytes transferred as well as the transport protocol. These informations are then used to generate a library of graphlets that are used to seek for matches and classifying captured traffic. The proposed approach copes with all the existing privacy, technical and practical constraints that prevent the usage of payload-based approaches. It was tested against three real traffic traces captured in the Internet link of two access networks. BLINC was able to classify more than 90% of all flows with more than 95% accuracy. However, connection patterns require a large amount of flow data and finished flow lifetime to perform the analyses. BLINC is then more suitable for an off-line traffic analysis of multiple flows.

The use of Bayesian classification methodologies was proposed in [MZ05], where hand-classified traffic data was used as input to a supervised Naïve Bayes classifier. The discriminators used for this analysis included the TCP ports, the Inter-Arrival Time (IAT) and its Fourier transform, the payload and the effective bandwidth. By performing some refinements over the classifier, the authors were able to reach an accuracy of 95% when mapping traffic of several protocols into different categories, as shown in table 2.3. The main disadvantage of this approach is that requires many training traces, since the ratio between training and test traces is 1:1, which is not always achievable. In addition, if the network and/or traffic parameters change, the classifiers have to be re-trained.

In [BTA+06], the early identification of TCP traffic was addressed based on the analysis of the first five packets of a TCP connection. The authors collected the size of these packets and used *unsupervised clustering* techniques to group the packets presenting similar profiles. A training phase was used for creating the classes, while the classification phase uses them to determine the application associated to each TCP flow. Traffic belonging to several protocols was accurately identified, with accuracy rates always higher than 80%. This work

Table 2.3: Network traffic allocated to each category in [MZ05].

Classification	Example Application
BULK	FTP
DATABASE	postgres, sqlnet oracle, ingres
INTERACTIVE	ssh, klogin, rlogin, telnet
MAIL	imap, pop2/3, smtp
SERVICES	X11, dns, ident, ldap, ntp
WWW	www
P2P	KaZaA, BitTorrent, GnuTella
ATTACK	Internet worm and virus attacks
GAMES	Half-Life
MULTIMEDIA	Windows Media Player, Real

was of seminal importance since it opened a wide range of new possibilities for on-line traffic classification. However, several issues are associated to this approach. To begin with, the approach is sensitive to packets arriving out of order, since the spatial representation of the traffic flows changes. In addition, applications exchanging similar packets will be assigned to the same label, while applications presenting unknown behaviors are not classified. Finally, this approach does not cope with traffic encryption that may prevent the analysis of the TCP headers.

In the same year, the work published in [EAM06] demonstrated that cluster analysis can be effectively used to identify groups of traffic that are similar using only transport layer statistics. Two unsupervised clustering techniques (K-Means and DBSCAN) were used to achieve an accurate traffic identification. The first was chosen due to its simplicity and works by partitioning objects into K number of clusters. The algorithm starts by randomly choosing the K centroids of each cluster and assigns each observation to the closest cluster. The K centroids are then recomputed and the assignment of observations to the closest cluster is repeated. This process is repeated until a convergence criteria is met. Such criteria can include no (or minimal) reassignment of observations to new cluster centers, or minimal decrease in squared error [JMF99]. DBSCAN (Density Based Spatial Clustering of Applications with Noise) is a clustering algorithm that considers clusters as dense areas of objects separated by less dense areas [EpKSX96]. Therefore, the created clusters can have an arbitrary shape, being not limited to a spherical one, and does not require a pre-defined number of clusters, which can be seen as an advantage over partition-based algorithms. The third used algorithm was AutoClass, which is a probabilistic model-based clustering technique and allows the automatic selection of the number of clusters. The defined clusters allow the objects to be fractionally assigned to more than one cluster. The probabilistic model is then built by determining the number of clusters and by inferring the parameters of the different probabilistic distributions. The studied protocols were HTTP, P2P, POP3 and SMTP and the mentioned clustering techniques were used, comparing their accuracy. The connections that DBSCAN labeled as noise reduced the overall accuracy of this algorithm, since they are considered as misclassification mistakes. However, DBSCAN presented the highest accuracy when classifying three of the studied protocols, while K-Means was the fastest approach.

Instead of classifying traffic based on statistics of individual flows, the authors in [HCL09] focused on building behavioral profiles describing the dominant patterns of a target application. A two-level matching mechanism is then used to classify captured traffic, where the first determines if a host participates in the application by comparing its behavior with the profiles. Subsequently, each flow of the host is compared to the profiles in order to identify the ones that were generated by the studied application. The selected target application was P2P and several rules were obtained for TCP and UDP connections, which are merged from different training traces. Then, the authors looked back at the behavior of each host

to construct application profiles. The classification results proved that their approach can accurately identify BitTorrent traffic. However, the number of rules required for classification was very high, which raises some issues on the scalability of their approach as well its ability to classify traffic on-the-fly.

In a recent work [HGPS11], the authors propose a "two-way" application of k-means clustering techniques that consists in analyzing a bidirectional flow as two unidirectional flows. The authors argue that, in this way, they are able to increase the classification accuracy by as much as 18% when compared to other similar approaches. In addition, they state that their approach generates fewer clusters, which implies that fewer calculations have to be performed to classify traffic. Several discriminators were proposed and the authors used their own version of the Sequential Forward Selection (SFS) algorithm to choose the best discriminators. It starts by clustering the training data according to each of the several discriminators separately and then determines the best ones by evaluating how many flows were assigned to the correct cluster. In the following iterations, the previously selected discriminators are combined with all the others individually to cluster the data. The best combination is selected until no improvement is made. The k-means clustering technique was used due to its fast training times and ease of implementation. Their results showed indeed an increase in the accuracy when compared to some other works.

However, this method suffers from the fact that traffic with the same statistical behavior can be classified as belonging to the same application, which may not be true, and traffic with unknown behavior is not classified. Clustering techniques are useful tools for grouping traffic with similar characteristics [EAM06], but they have to rely on other identification techniques to label the clusters. Machine Learning Classifiers are also based on the statistical analysis of Internet traffic and can provide accurate identification results using only transport layer information [MHLB]. Recent works use wavelets, alone or combined with other techniques, on the detection of traffic anomalies [GHYC06, LG09]. However, these works did not explore the multi-scale characteristics of network traffic for the detection of network anomalies and do not provide a flow-based analysis, which is more suitable for the classification of attacks.

2.2.4 Real-Time Classification Approaches

Real-Time traffic classification is a fundamental task for many network management decisions: by timely identifying the applications that generate traffic on a specific network link, network managers can optimize the utilization of their networks; better Quality-of-Service (QoS) can be offered to connected clients, while preventing the saturation of many network resources; the timely identification of malicious traffic or traffic presenting anomalous patterns is also crucial to assure the protection of the connected hosts and network resources. However, achieving such ability is not an easy task. The inherent complexity of current network applications and services and the existence of several privacy and legal restrictions that prevent

the analysis of the packets contents are important obstacles for an accurate and timely traffic classification. One of the first research works that addressed this issue [NA06] emphasized the need to achieve an accurate traffic mapping well before a flow has finished, also considering situations when the onset of the flow was lost. Therefore, this reference proposed a training approach for the classifier based on short sub-flows, extracted from full-flows examples of the studied Internet applications. This optimization was subsequently evaluated by deploying a Naïve Bayes classifier, achieving a high classification performance.

In 2007, a work presented in [BMM+07] studied the real-time detection of Skype traffic. The authors presented a model for the Skype message building process and described in detail the several messages exchanged by Skype clients. The approach consisted in two complementary techniques: the first uses the Pearson's Chi-Square statistical test to detect whether and which messages are encrypted and to detect Skype's fingerprint. This allows the distinction between the traffic generated by Skype clients from the one of other VoIP sources. The second builds stochastic models based on packet arrival rate and packet length, which are then used as features on a decision process based on Naïve Bayes classifiers. This allows the quantitative evaluation of the resemblance of potential Skype flows to the characteristics inferred from the stochastic models. The accuracy of the obtained results was verified by comparing the results obtained with these two presented classifiers with the ones obtained when performing payload inspection. Note that this inspection is made difficult by obfuscation and cryptographic techniques. When using both presented techniques, the percentage of False Positives drops to almost zero.

In [HJC08], a set of flow attributes is proposed to characterize the negotiation behaviors, in the application layer perspective, for both TCP and UDP traffic flows. The authors state that an application-layer perspective shows more potential discriminating characteristics than a transport-layer perspective. In addition, the authors also state that these attributes are available at the early-stage and consequently, are suitable for real-time traffic classification. The authors defined application interaction rounds as the basic "block" for capturing the application characteristics, where each one consists of two parts. These parts consist of two TALK blocks, where the first is a series of data packets transmitted in one direction and the second is a series of data packets transmitted in the opposite direction. TCP control packets were used to determine the initializer and listener of each flow. For UDP flows, the initializer is the one that sends the first packet. Several discriminators such as layer 7 transmitted size, throughput, IAT and response time were defined and analyzed for the different defined TALK blocks. Several machine learning algorithms implemented in WEKA [WEK11], such as Naïve Bayes, Sequential Minimal Optimization (SOM) and pruned C4.5 decision trees, were deployed for assessing the accuracy of the discriminators. The classification accuracy using all these approaches was determined and proved that, using their ML approaches with the application layer metrics, the authors could achieve a high accuracy with low False Positive

(FP) rate. The authors also claimed that this work presented an accuracy increase of around 8% to 21% when compared to some other previous works. However, this work lacks of novel classification methodologies, since the authors only deploy machine learning algorithms implemented in WEKA and their approach is not able to analyze encrypted traffic.

In a more recent work [JG10], authors propose a FPGA-based parallel architecture to accelerate the statistical identification of multimedia applications, while assuring high accuracy. Applications such as Skype, Instant Messaging and IPTV were studied and by using the k-Nearest Neighbors (k-NN) algorithm and a Locality Sensitive Hashing (LSH) the approach can be deployed in high bandwidth links. According to the presented results, this approach was able to achieve an accuracy of more than 99%. In [BBL10], the fast identification of BitTorrent traffic was addressed by using machine learning techniques to select the features that could be used for real-time traffic classification. The importance of a timely identification of BitTorrent traffic comes from the fact that it is the most popular P2P client and is one of the most dominant traffic generating applications in the Internet. Therefore, its early identification allows network operators to better manage the resources of their networks and provide a better QoS to their clients. The authors examined complete TCP flows in order to determine the best differentiating statistics for BitTorrent classification, and four discriminators were proposed:

1. *Minimum payload*: some messages exchanged by the BitTorrent protocol present small size packets, with 5-17 bytes of payload;

2. *Small Packet Ratio*: BitTorrent protocol serves two purposes: data exchange and information updates between peers. *Small Packet Ratio* is then defined as the ratio of the count of small packets to total packets within a flow;

3. *Large Packet Ratio*: This is defined as the the ratio of the count of large packets to total packets within a flow;

4. *Smaller Payload Standard Deviation*: Each flow consists of data flowing in two directions. For each direction, the standard deviation of the TCP payload size was computed and the smallest value was used.

The suitability of the presented discriminators was then evaluated using sub-flows with 300 packets size, in order to assure that each has at least one characteristic packet, using the approach presented in [NA06]. The authors have trained and tested a classifier based on the C4.5 algorithm implemented in WEKA and evaluated the four discriminators separately and together. In addition, sub-flows of different sizes were used to evaluate the effect of the number of analyzed packets in the classification accuracy. Accurate classification results were obtained when using the four discriminators and when evaluating flows of 150 and 300 packets. In addition, the authors were also able to distinguish other client-server bulk transfers from

BitTorrent traffic. However, their approach relies on the analysis of TCP packets and it is known that there are implementations of BitTorrent protocol that use UDP as the transport-layer protocol, which prevents the classification of such traffic.

2.2.5 Anomaly Detection

The problem of anomaly detection consists in identifying, or finding, patterns in data that are compliant with the expected behavior. Anomaly Detection can be used in many different fields; in network security, it is more related to dealing with several security issues, like identifying patterns of unknown attacks, intrusion attempts or zero-day attacks. Different approaches have been proposed to face this diversity of problems.

Machine Learning

In [LCD04], a methodology for the detection, identification and quantification of *network-wide* anomalies was proposed. Principal Component Analysis (PCA) was used to perform the separation of the high-dimensional space occupied by a set of network traffic measurements into disjoint subspaces. Such subspaces correspond to normal and anomalous network conditions, enabling the identification of volume anomalies and their corresponding flows. The authors analyzed data collected from two backbone networks, Sprint-Europe and Abilene, and deployed PCA on a measurement matrix that denotes the time-series of all links. They determined that although both networks have more than 40 links, the vast majority of the variance in the time-series of each link can be captured by 3 or 4 principal components. These components are then mapped into new axis, which are then separated in two subspaces: the *normal (S)* and *anomalous ($\overline{S}$)* subspaces. Anomalies are then detected by separating the link traffic into their normal and anomalous components. To evaluate their approach, the authors first isolated true anomalies and evaluated their subspace method quantitatively, more precisely the detection probability and the false alarm probability. Then, anomalies of different sizes were inserted in different flows and the proposed detection approach was applied. The detection rates obtained were very high and are independent of when the anomaly was injected.

Another work [WZ06] proposed the use of clustering algorithms together with factor analysis and Mahalanobis distance. Factor analysis allowed the authors to uncover the structure of a set of variables from an unknown sample, reducing the attribute space to a smaller number of factors, which enables a more efficient characterization of normal activities. The Mahalanobis distance was used for determining the "similarity" between a set of values extracted from an unknown sample to a set of values extracted from known samples and, therefore, determining if those unknown samples constitute, or not, an anomaly. The authors were then able to (i) identify outliers based on a training model and (ii) cluster attacks by abnormal features. The 1998 DARPA intrusion detection dataset [LFG+00] was used for obtaining attack-free data

for the training set and the 1999 Dataset [Dar11] was then used to evaluate their approach. The experimental results showed that the proposed approach is able to accurately identify Internet attacks with a tolerable false alarm rate.

In [SM07], the authors propose a general framework for the detection and classification of novel attacks that uses a new Support Vector Machine (SVM) technique, named *enhanced* SVM, which combines supervised and unsupervised SVM techniques for providing unsupervised learning and low false alarm rate. It consists of an hybrid machine learning approach for anomaly detection. The overall structure of the framework comprises four major phases/components and is depicted in figure 2.2. The first consists on the on-line processing of captured traffic and a real-time filter using TCP/IP Fingerprinting is used to drop malformed packets. An off-line processing is also performed in this phase and includes data clustering using Self-Organized Feature Maps (SOFM), which is an unsupervised neural network model for analyzing and visualizing high-dimensional data into two dimensional lattices in order to create a profile of the normal and legitimate traffic. In addition, a packet field selection using Genetic Algorithms (GA) is also performed in order to extract optimized information from raw Internet packets. The subsequent phase consists of preprocessing the filtered packets in order to allow a high detection performance. Packets relationships based on traffic flows are considered in order to charge Support Vector Machine (SVM) with temporal characteristics, during this phase. The third phase consists in training/testing the previously mentioned Enhanced SVM, which combines two learning methods: soft margin SVM (supervised method) and one-class SVM (unsupervised method). In this manner, the Enhanced SVM inherits the high performance of soft margin SVM, while presenting a high novelty detection capability associated to one-class SVM. Finally, the last phase consists in verifying the approach using a validation test that showed that the proposed approach managed to detect novel forms of attacks, while presenting a low False Positive rate.

A more recent work [DHKR09] proposed the use of Machine Learning (ML) techniques to explore the correlations between packet and flow level informations, allowing the association of packet level alarms with a feature vector inferred from flow records. The authors claim that their work presents some key contributions such as the ability of detecting unwanted traffic using a set of flow signatures, which allows their classifiers to operate in network links with very high-speed links. A set of *flow level predicates* is built from each flow indicating its transport-layer protocol and other numerical attributes such as the number of packets. The rules from the popular IDS Snort [Sno11] were used, and for each one a score was computed over the previously mentioned flow attributes. If this score exceeds an operating threshold θ, a ML alarm is issued while classification mistakes are minimized by assigning weights to each Snort rule. An architecture for exploring their methodologies at a network scale was also proposed, where flow records are collected from a set of interfaces across the monitored network in order to capture all flowing network traffic. A set of packet monitors is then

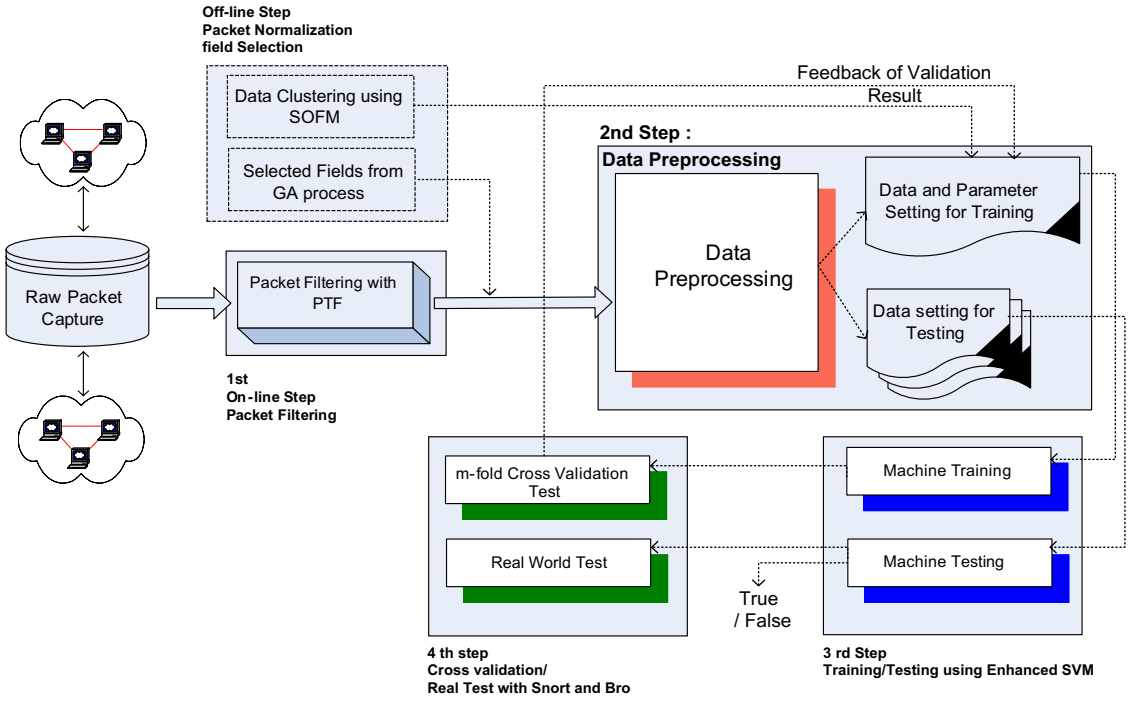

Figure 2.2: Flow Diagram for the anomaly detection approach proposed in [SM07].

deployed in specific sites in order to capture a representative mix of traffic, and each monitor is equipped with a set of packet level rules. The produced alerts are then forwarded to the ML trainer that correlates the different packet alerts with the flow generated by the same traffic and generates a set of flow level alerting rules. Finally, a *Runtime Flow Classifier* applies these rules to all flow records, producing flow-level alerts.

In [BSM10], an analysis of the effects of packet sampling and of temporal aggregation on the signal properties was carried out and some techniques were suggested to overcome such effects. The authors state that these techniques introduce noise, distortion and aliasing on the analyzed signal and show that the aliasing introduced by the aggregation step has the largest impact. They subsequently propose a replacement for this step, which consists on the implementation of a low-pass filter that decreases the aliasing effects. The authors proved that, by applying their approach, the performance of anomaly detection systems can be largely improved.

Wavelet Analysis

Wavelet analysis has been widely applied on the field of anomaly detection due to its time-frequency property that allows the decomposition of a signal into several components, each one at a different frequency.

In [KPA02], wavelets are used for the analysis and characterization of flow based traffic behaviors. NetFlow signals are split into three different frequency ranges, starting from the low frequency components that account for patterns over long periods of time, such as several days. Mid-frequency components account for daily events such as daily variations in the flows attributes, while high-frequency components account for short term variations. These three components are obtained by dividing the obtained wavelet decomposition coefficients in three different intervals, generating signals from these coefficients. Subsequently, algorithms are used to detect anomalies on the generated signals by setting thresholds at the mentioned levels. The obtained results showed that this approach was able to provide an accurate identification of some forms of DoS attacks and port scans due to the anomalies that are generated in the mid and high-frequency components. Nevertheless, some recent and stealth attacks do not generate anomalies at this frequency levels, and consequently, cannot be detected using this approach.

Kim *et. al* proposed in 2004 a technique for the detection of traffic anomalies by analyzing the correlation of the destination IP addresses in the traffic flowing from a local network [KRV04]. Their study was performed at an egress router and the authors stated that destination IP addresses presented a high correlation degree and changes in these values could indicate an anomaly. DWTs were then applied to the correlation data of the destination addresses over several time-scales. Deviation from normal profiles generated alarms that could be passed to network administrators.

A work addressing some of the limitations of wavelet based anomaly detection, such as its computation complexity, was presented in [GHYC06]. The authors presented a novel anomaly detection approach based on wavelet transforms that is able to adjust the decomposition process adaptively. The algorithm can select different time-frequency resolutions according to the characteristics of the analyzed traffic signal. It maintains the same ability of detecting anomalies at various frequencies, specially at the mid-range and high frequency components that cannot be detected by multi-resolution analysis. The results obtained with simulated attacks showed that the proposed approach could detect network traffic anomalies in a timely manner. However, DDoS was the only simulated attack, so other stealth forms of attacks should have also been simulated in order to verify the ability of the proposed approach to detect them.

In [KR06], the authors present *NetViewer*, which is a network measurement approach for the real-time detection, identification and visualization of security attacks and anomalous traffic. *NetViewer* passively monitors and extracts samples of network packet headers that are then represented as images. Therefore, a series of samples can be seen as a sequence of frames or video, enabling a visual representation of the attacks that can be understood by the human eye. Some image processing and video compression techniques can be used for the detection of anomalies. The authors state that "scenes changes" can reveal sudden changes

in the captured traffic or anomalies and that "motion prediction" techniques can be used to understand some of the attacks. *NetViewer* is composed by three components. The first consists in generating the image signal from the samples of network traffic. The second consists in detecting the anomalies that are present in the generated images. As already mentioned, image processing and video compression techniques are deployed at this phase. Finally, in the last stage the detected anomalies are identified together with the identification of the attacker and victim. The obtained results were very promising, with an overall accuracy of more than 90%. However, this analysis approach does not seem to cope with traffic encryption since the packet headers may not be accessible.

In a more recent work [LG09], the authors proposed a new network signal modeling approach where wavelet approximations are combined with system identification theory. The architecture of the proposed approach consists of three components:

1. *feature analysis*;
2. *normal network traffic modeling*;
3. *intrusion detection.*

In the first component, the authors defined and generated fifteen features to characterize network traffic behaviors. Based on these features, the second component models and represents normal daily traffic by a set of wavelet approximation coefficients, which can be predicted using an AutoRegressive with eXogenous (ARX) model. The output of this model represents the deviation of the input signal from the normal behavior signals and is then passed to the third component that performs intrusion detection by using an algorithm to detect outliers. The 1999 DARPA intrusion detection dataset [Dar11] was used to validate this identification approach and the authors were able to accurately identify different types of attacks. In addition, the accuracy of the approach was also verified by three days of collected traffic from Fred-eZone, a free Wireless Fidelity (WiFi) network service provider [Fre11] and the results were still accurate although some attacks were not detected, which decreased the classification accuracy. As a drawback, we can mention the fact that the data from which the normal traffic models are inferred needs to be free of intrusions and attacks, which is not easy to guarantee.

Wavelets have also been applied, with promising results, on the implementation of prototypes for the detection of specific types of attacks. For instance, in [Ram02] an approach named WADeS (Wavelet based Attack Detection Signatures) was proposed for the detection of DDoS attacks and consisted in applying WTs on captured network traffic signals. Subsequently, the variance of the decomposition coefficients was used to estimate the occurrence of the attack. Another prototype for the real-time detection of anomalies, named Waveman, was proposed in [HTS06]: authors used different metrics to evaluate the performance of various

wavelet functions on detecting different types of anomalies, like DoS and port-scans, part of the 1999 intrusion detection dataset [Dar11] and also real network traffic data.

2.3 Intrusion and Attacks Detection

As mentioned in chapter 1, the recent and stunning increase in the number and variety of Internet attacks has given a tremendous importance to the network security area. Several solutions are currently being deployed, at a network or an host level, for the identification, mitigation and prevention of IP-based attacks. In the following sub-sections, both approaches will be presented, together with the associated advantages and drawbacks.

2.3.1 Host Level Solutions

Host based approaches reside in the monitored host and aim to monitor activities and events of a single host, tracking changes that were made to important files and directories in order to detect suspicious activities. The data sources comprise system and processes logs, file system monitoring, network configuration monitoring and many more.

One example of host based approaches are *anti-virus* applications, which rely on a database containing known patterns of attacks that enables an accurate identification of those attacks. Such databases need to be constantly updated in order to detect new and emerging threats and, with the ever increasing number of new attacks and vulnerabilities, can become very complex and unmanageable. In addition, generating the signature for the identification of an unknown threat can be a time-consuming and complex task due to the amount of traffic samples that must be analyzed in order to generate the correct signature [YA09]. Anti-virus are also unable to discover highly sophisticated and stealth attacks that present unknown patterns or patterns similar to legitimate applications. In addition, once a machine becomes infected, these tools are not able to detect the illicit traffic it sends, which may consist of requests to servers or other hosts on the network or traffic containing stolen confidential data. All these communications present profiles that are similar to normal traffic and, consequently, are not detected as illicit traffic. These are the biggest threats to corporations because, once a host becomes infected, the whole network, its services and confidential data are compromised.

Personal *firewalls* can also be an effective tool for preventing a computer infection by blocking traffic from unauthorized applications. This protection can be compared to a "digital shield" around the host that restricts incoming and outgoing network activity to the monitored host. Although it can provide a considerable protection, by blocking some illicit applications, *firewalls* require an average know-how from the user, which rarely happens: by misusing the firewall, an user can inadvertently open a breach in his network security. Moreover, personal firewalls cannot prevent illicit traffic embedded in normal communications supported by authorized applications or services.

There are several issues associated to host-based and monitoring approaches that have already been described and discussed. However, an important aspect is the fact that the ever-growing databases used by Host Level Solutions force these softwares to consume more and more resources of the protected host, which constitutes a major drawback [YA09]. In addition, host based approaches require that one agent per monitored host is deployed, which raises the cost of deploying such approaches on a large network.

2.3.2 Network Level Solutions

Network based detection approaches monitor specific network segments and devices and analyze the flowing network traffic for suspicious patterns and activities. At this level, IDSes such as Snort [Sno11, Roe99] and Bro [Bro11], and network *firewalls* may appear as the most adequate approaches for guaranteeing the security of a network, its connected hosts and its infrastructure. These systems are deployed in strategic points of the network, such as ingress and egress nodes, so that all the traffic flowing to and from all hosts of the monitored network can be analyzed.

IDSes operate by inspecting the contents of the captured packets in order to find digital signatures, or patterns, of known threats. Such patterns are stored on a database that needs to be constantly updated. However, scanning every single packet and inspecting its contents, against the databases that contain all known attack patterns, is a complex and computationally intensive task. This raises several scalability issues that prevent their deployment on network links with high bandwidth since these systems will not be able to analyze and compare the contents of all packets flowing through such links. IDSes can also be deployed in a distributed manner, in which several probes monitor each one of the network hosts. However, the correlations that the probes must perform in order to discover distributed and stealth attacks are extremely complex. Since IDSes also rely on databases containing the known forms of attacks, they suffer from the same drawbacks associated to anti-virus, which were already listed in the previous sub-section. These include the need of constant updates and an increase on the complexity of such databases and the inability of detecting unknown threats and stealth forms of attacks presenting normal traffic characteristics [KDL04]. As pointed out in [Gol11], some simple techniques can also be used to circumvent detection by IDSes. Finally, these tools are also unable to cope with encrypted traffic and with the diverse confidentiality restrictions, all of them preventing the analysis of the contents of the packets.

Network firewalls scan the flowing network traffic and use a set of rules to decide which traffic can pass through the firewall and which traffic must be dropped. In this manner, traffic generated by illicit or suspicious applications can be easily blocked. Firewalls usually create a Demilitarized Zone (DMZ), which separate servers from the remaining protected network in order to avoid the propagation of intrusions. However, several issues are also tipically associated to network firewalls, including their inability to offer a secure protection against

stealth attacks targeting authorized services on the monitored network.

2.3.3 Hybrid Approaches

Cisco's Intrusion Prevention System (IPS) is an example of an hybrid platform for the detection of several types of attacks and threats that works simultaneously at the network and host levels [Cis11]. Threats are mitigated by performing deep packet inspection of all network packets and monitoring the processes that are running on the different network hosts. Such software usually limits the user ability to run other types of programs or perform tasks different from the ones that are allowed by the monitoring software. This fact usually dissuades companies from using this platform. Besides, since the platform relies on an extensive database of known threats and attacks, it suffers from the same limitations of all other IDSes.

2.3.4 Conclusions

We can then conclude that there is a stringent need for new methodologies that can identify unknown, distributed and stealth attacks, can cope with different privacy restrictions that govern most of current networks and with traffic encryption. In order to achieve these requirements, new paradigms for the analysis of IP traffic and for the identification of IP-based security attacks must be developed.

2.4 Botnets

The recent and alarming increase on the number, variety and stealthiness of reported Internet attacks is tightly connected to the emergence of *Botnets*, which are nowadays considered the most serious threat to the Internet. They have become the cornerstone of on-line criminal activities and consist of networks of compromised hosts running an autonomous piece of software, the *bots*, and controlled via a command and control (C&C) infrastructure [Ell10]. The administrator of this infrastructure is known as as the *bot-master* and uses it to send instructions and commands to the controlled machines. So, a botnet consists of a collection of compromised machines running the bot program under the control of the *bot-master*. Botnets are used for several illegal purposes that will be presented and discussed in sub-section 2.4.3.

Botnets recruit new vulnerable systems using methods also deployed by other classes of *malware*, such as remote exploitation of known vulnerabilities. However, what distinguishes them from the remaining classes of malware is the fact that botnets use a communications infrastructure and the fact all compromised machines are able to cooperate towards a single purpose.

2.4.1 Command & Control

Communications betweens *bots* and their *masters* are achieved through a C&C infrastructure. The interaction of the compromised hosts (*bots*) with this infrastructure and, consequently, with the *bot-master* is depicted in figure 2.3. The *bots*, under the control of the *bot-master*, connect using the pre-defined communications channel to obtain the instructions. This channel can be implemented using a variety of Internet protocols, including the ubiquitous HTTP protocol, P2P networks or the IRC (Internet Relay Chat) protocol, which is still widely used. In fact, in an early stage, such communications were performed using the IRC protocol, since it was a scalable solution that required minimal administration efforts [CJM05]. The simplicity of the IRC protocol and its flexibility in allowing different forms of communications (point-to-point, point to multi-point) are some of the reasons why *bot-masters* still prefer this communication protocol. In addition, its flexibility and the availability of many open-source implementations, allowing *bot-masters* to use their own version of the protocol, are also very important factors [ARZMT06] [NI07]. However, blocking such communications was very easy to achieve by simply closing the ports used by the IRC protocol. Consequently, the structure of *botnets* had to evolve to a more distributed one, involving Peer-to-Peer architectures and protocols [DD08, WSZ10]. In fact, the main feature a P2P C&C is the fact that there is no centralized server that can be shutdown, which makes the disruption of this infrastructure much more difficult. On the other hand, as already mentioned, these communications can run on top of the HTTP protocol, whose ports are never closed in private networks, making this infrastructure very resilient. C&C communications can also be hidden inside common traffic patterns or encrypted traffic, making their detection and disruption even more difficult.

The *bot-master* can then send his instructions to the compromised hosts using the communications channel. In this way, the compromised hosts can be controlled remotely without the knowledge of the owners of the compromised machines. These instructions can include scanning for vulnerable systems or taking part in a distributed attack.

2.4.2 Botnets Life Cycle

Botnets follow similar steps throughout their existence. The general life cycle of a botnet can be divided in four main phases (figure 2.4):

1. initial infection - exploit;
2. secondary infection - bot download;
3. maintenance & update - join;
4. malicious activities - commands.

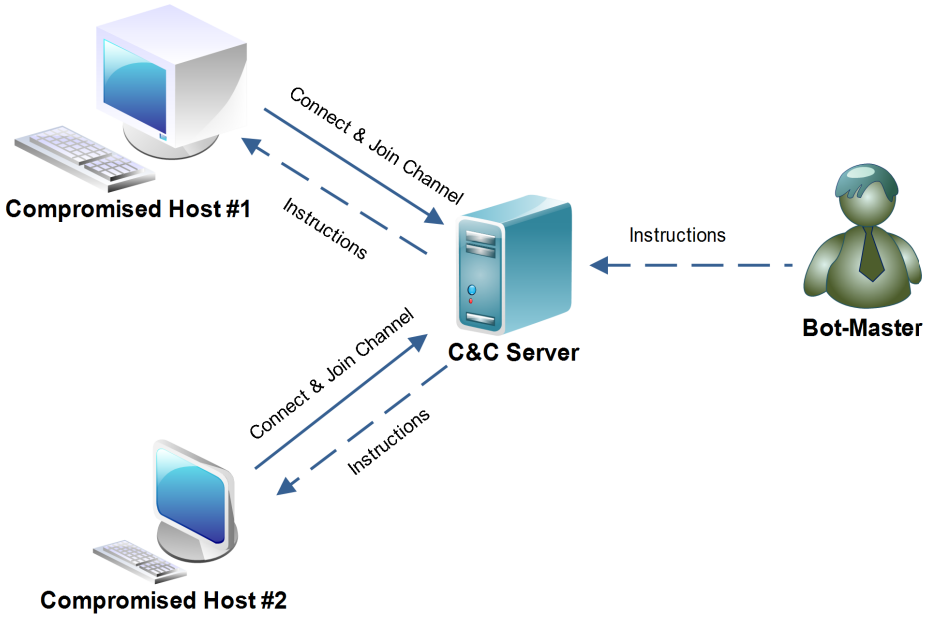

Figure 2.3: Communications between compromised hosts and the *bot*-master using the C&C infrastructure.

The first phase comprises the initial infection of a machine that can be performed in many different ways, including exploiting the vulnerabilities of an operative system or software. Host scanning is one of most used techniques for assessing the open ports, and consequently, the open service on a specific host. In addition, users may perform an accidental download and execution of malicious code while opening e-mail attachments or browsing through compromised or malicious web-sites. After the initial infection, the second phase (secondary infection) takes place, consisting in the download and execution of the botnet code so that the compromised machine can join the botnet and perform the requested actions. This download can be performed by using several protocols such as FTP, TFTP and HTTP. Subsequently, a secure connection with the C&C server must be performed and the bot-master has to obtain feedbacks from the bots, perform updates and/or add modules to the malicious code running in the different machines before performing an attack. Finally, the fourth phase consists in performing the attacks ordered by the bot-master. In the following sub-section, the most relevant usages and security attacks of botnets are presented and discussed.

2.4.3 *Botnet* Uses

The cumulative and seemingly infinite computational power and bandwidth resources of all compromised hosts make *botnets* suitable for performing highly distributed, stealth and massive attacks [SG10, ADPG+10, Ell10]. In the following paragraphs we present and explain some of the usually conducted security attacks.

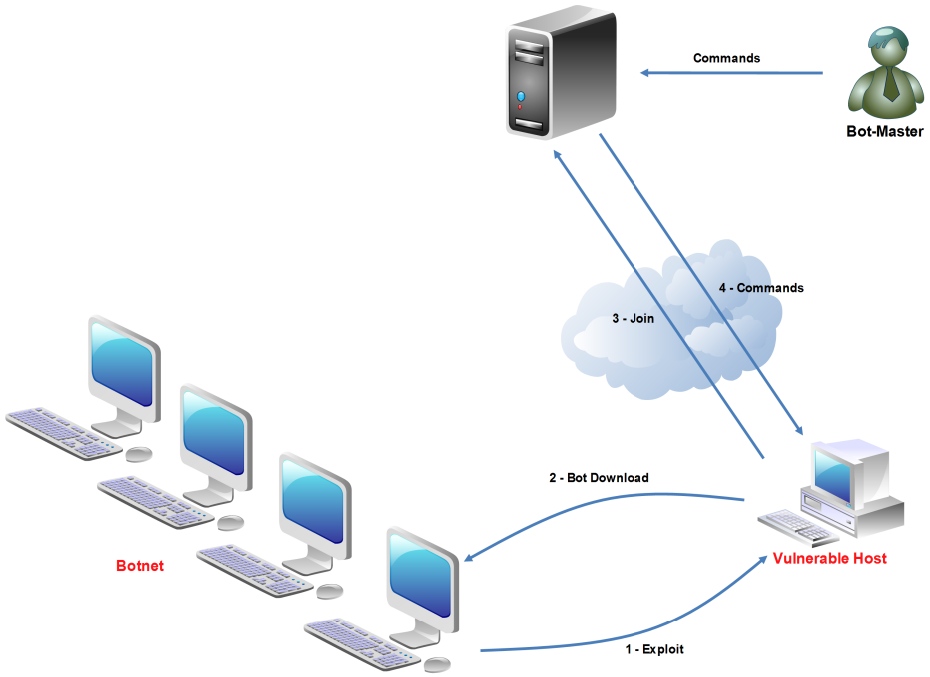

Figure 2.4: Botnets life-cycle

Distributed Denial-of-Service (DDoS) attacks

The idea behind a Denial-of-Service (Dos) security attack is to make a computer resource unavailable to its users. Most commonly, such computer resource consists of an Internet site or service and the attack is an attempt to prevent its correct functioning or its availability. So, a DDoS consists in a multitude of compromised systems attacking a single target and *botnets* are specially tailored for this type of security attacks. Some of the methods for these attacks use UDP and SYN flooding [NI07].

Sending Spam

One of the most popular uses of Botnets is spamming, which consists on sending unsolicited e-mail messages. According to a recent report [Lab10], in the fourth quarter of 2010 more than 50 billion spam messages were generated per day. The same report states that spam was responsible for more than 80% of all e-mails.

Botnets have enabled such type of attacks to be deployed at large scale due to the combined bandwidth and computational power of all compromised hosts under the control of a bot-master. Many other advantages can be enumerated, such as shifting all costs (computational, bandwidth and reputation) of performing such attack to the true owners of the bots. In addition, the use of the IP addresses of the bots prevents counter-measures such as blacklisting

IP addresses originating spam messages.

Flooding Attacks

These attacks target at saturating the available bandwidth in a network. They work by sending a large volume of traffic consuming all bandwidth of a connection or overwhelming the resources of routers and servers.

Exploit Scanning

Another usage of *bots* is to perform host scans in order to determine the open ports and their services in the surrounding systems. In this manner, the vulnerabilities associated to such services can be used to gain control of those machines.

Download and installation

A common feature all *bots* present is the ability to download and execute binaries. Such downloads can be performed by FTP, TFTP and HTTP and are the most used method for updating the malicious code in a compromised host.

Click Fraud

A click fraud occurs when an automated script or a computer program attempts to imitate a legitimate user performing a click on *pay-per-click* advertisement [MKR07]. *Botnets* are the perfect tool for carrying out such type of fraud since they are composed by hundreds/thousands of compromised hosts that can issue web requests representing "clicks" on the previously mentioned advertisements. These frauds are very difficult to detect, since they are carried out by different hosts in different locations [NI07].

Phishing

Recently, phishing attacks have increased dramatically and are one of the main causes is the fact that botnet malware started to incorporate phishing abilities. These consist in displaying pre-built fake web-pages visited by Internet users or redirecting the user to a fake web-site controlled by the *bot-master* in order steal authentication credentials [DCJ10] [Sec10]. Such events are usually triggered by keywords in the address of the visited web-page, which correspond to sites known by hosting financial and on-line banking services.

2.4.4 Detecting *Botnets*

The detection of *botnets*, their compromised hosts and attacks is a complex task. To begin with, the traffic *bots* exchange resembles legitimate communications. In addition, since these

networks are composed by several thousands of infected machines, their attacks are highly distributed. IDSes, such as Snort [Sno11] and OSSEC [OSS10], may seem appropriate tools for such task but their inability to detect *zero-day* threats and distributed attacks prevents them from efficiently detecting *botnets* attacks [BY06]. In some cases, communications between the compromised hosts and the *bot*-master are encrypted, making most of the IDSes unable to detect them. In fact, *botnets* have become so ubiquitous that the Shadowserver Foundation continuously monitored more than 6000 *botnets* C&C servers in 2010 [Sha11]. Nowadays, more than 100 million computers are estimated to be part of criminal networks.

There are three main *botnet* detection approaches: active and passive approaches and approaches based on *darknets/honeynets* [SG10]. Active approaches imply capturing *bots malware*, deactivating its malicious parts and analyzing all the commands sent and received while executing the harmless bot code. In addition, specialized *bots* that simulate the behavior of real *bots* can also be created. This enables the *bot* to connect to a C&C server and observe the activity on the mentioned server. The main drawback of this approach is that such *bots* are easily detected by the *bot-masters* and, consequently, several counter-measures can be taken, such as disconnecting the mentioned *bot*. On the other hand, passive approaches work with more subtle sources of information, such as the traffic generated by the analyzed *botnet* and other created effects like broken packets and uncompleted sessions. Since they do not create any flow of information back to the *botnet* control infrastructure, they are not detected by the *bot-master*. Finally, *darknets* constitute a completely passive approach because they are composed by Internet systems used with the only purpose of being compromised, thus allowing an insight into the studied Internet threats. Such approach provides far more critical information than any other available security tool [AH10]. Similar to network telescopes, *darknets* are deployed in unused IP addresses [SG10, YBP05]. In the following sub-sections we present these approaches, together with the most relevant work that has been done so far.

Active & Passive Analysis

Several approaches have been suggested for addressing the *botnet* detection problem. Based on the fact that *botmasters* perform DNS blacklist (DNSBL) queries to determine if their spamming *bots* are listed, the authors in [RFD06] propose the use of heuristics to determine the queries that are likely to be executed by *botmasters*. Subsequently, the authors built query graphs that allowed them to focus on subgraphs presenting a higher percentage of reconnaissance lookups. In this manner, an identification of likely *bots* can be achieved. In addition, some high-level results indicated that *botnets* are being used to perform DNSBL reconnaissance on behalf of bots in other *botnets* and that the distribution of these queries suggests that such activities can be detected in real-time. However, most of DNS blacklists only respond to queries issued by verified mailhosts, which makes this approach inappropriate for the detection of compromised hosts.

In [DZL06], the use of time-zones for modeling the propagation of *botnets* is proposed. Authors base their work on the fact that most of the compromised hosts, being machines of Internet users, will be shutdown in night periods. Subsequently, diurnal propagation models were created using shaping functions, which allows the prediction of the *botnet* population growth. Consequently, the responses to the different *botnets* can be prioritized. However, the data collection process required by this approach is very disruptive and, consequently, can be easily detected by *botnet* operators.

In [KRH07], authors developed an anomaly-based passive analysis algorithm for the detection of IRC *botnet* controllers with less than 2% false-positives. Controllers running on any random port can be detected without using any known signatures. The proposed algorithm, shown in figure 2.5, uses transport-layer flow summary data, which reduces the amount of data that needs to be processed and identifies hosts with suspicious behaviors (suspected *bots*) by aggregating triggers. The flows sent and received by such hosts are then isolated and analyzed in order to identify candidate control flows. These control flows are then aggregated and analyzed in order to isolate suspected controllers and controller ports. The obtained results allowed the identification of 376 unique *botnet* controllers IP addresses between August 2006 and February 2007. Finally, the authors argue that their methods present several advantages, such as the fact that the data analysis is completely passive, so it cannot be detected by *botnet* operators and does not interfere with network operations. In addition, their analysis algorithm is scalable to very large networks and is able to show the dynamics of *botnets*. However, access to Tier-1 networks is difficult, which prevents the usage of this approach in general scenarios.

In the same year, a network quality indicator, named *uncleanliness*, was proposed in [CSF+07], which indicates the propensity of hosts in a network to be compromised by external entities. Authors state that unclean network will present two main properties, which are spatial and temporal *uncleanliness*. The first relates to the tendency for compromised hosts to cluster within unclean networks, while the second relates to the tendency for unclean networks to contain compromised hosts for long periods of time. Using reports of network activities and traffic logs of large networks, evidence of the previously mentioned properties were shown and such properties were then used to predict future *botnet* addresses.

In [YR08], the authors proposed a system to detect malware (including *botnets*) by aggregating traffic that shared the same external destination, similar payload and involved internal hosts with similar OS platforms. The main idea behind this approach is that malware rarely affects only one host in a network and, consequently, all the compromised hosts can be easily detected if such aggregation is performed to the traffic flowing to and from the gateway of the network. Binary vectors are formed for each one of the internal hosts, PCA is deployed for data reduction and clustering algorithms were used to group the vectors presenting similar behaviors. The authors were able to accurately detect platform-dependent malware infections

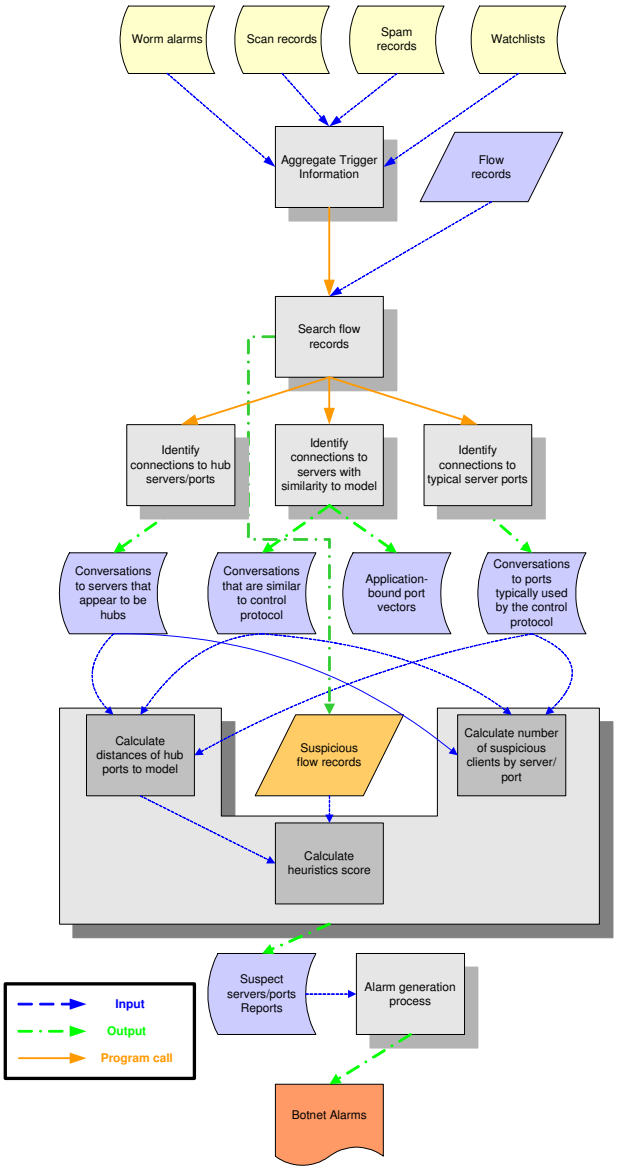

Figure 2.5: Flow Diagram for the *botnet* detection system proposed in [KRH07].

reporting to common sites. However, this classification approach does not cope with traffic encryption, since in these cases payload comparisons cannot be performed.

In 2009, a published work proposed a mechanism for detecting new types of *botnet* spamming attacks against Web e-mail providers through the construction of large user-user graphs and looking for connected subgraph components [ZXY+09]. In this manner, the authors can uncover and evaluate the correlations between *botnet* activities and detect stealthy *bots* that otherwise are difficult to detect in an isolated way. Botgraph was implemented as a distributed

application on a computer cluster in order to cope with the huge amount of data that had be analyzed and is composed by two components: aggressive sign-up detection and stealthy bot-user detection. The first consists in detecting sudden increases in signup activities from an IP address and a simple EWMA (Exponentially Weighted Moving Average) was used to accomplish such task. The second component consists in detecting stealthy-*botnet* users by building the afore mentioned user-user graphs. These are built by assigning to each user a vertex and the weight between two vertices is determined by the features used to determine the similarity between two users. Such features include the number of common IP addresses logged in by two users since, as the authors state, for each spammer the number of bot-users is much higher than the number of bots. This implies that multiple bot-users must log-in from a common bot, therefore, from a common IP address. The authors applied their detection tools to Hotmail logs containing informations from more than 500 million users, over which 26 million were successfully detected as *botnet*-created user accounts. In addition, their graphical approach proved to be adequate for the analysis of large datasets. Despite being a very promising and accurate procedure, it lacks the ability of identifying other types of *botnets* that do not generate spam but perform other types of security attacks, such as DDoS.

In a recent work [MGT+10], the analysis of the characteristics of packet size sequences belonging to TCP conversations between IRC zombies and their C&C servers was proposed. According to the authors, these conversations present a quasi-periodic nature, which allows their differentiation from the remaining TCP connections. For this purpose, the authors defined the Conversation Content Sequence (CCS) as the packet size sequence corresponding to the packets of the conversation between the IRC client and its server, after the client joins a certain channel. A framework was then developed for analyzing this traffic and detecting the flows generated by bot controlled machines. This platform starts by filtering IRC traffic from the remaining captured traffic and then computes the average packet size for all filtered flows. The ones exceeding a certain threshold are tagged as generated by *botnets*, while the quasi-periodicity of these flows is measured by determining the most frequent sub-string in the whole conversation. Consequently, the periodicity of each one of the flows is measured and, if it exceeds an established threshold, then the analyzed flow is considered as being generated by a *botnet*. The approach was tested in real *botnet* traces captured from honeynets and the authors reached an accurate identification.

Another recent work [BOB+10] performed a reverse-engineering of the Zeus botnet crimeware toolkit. This botnet was chosen since, in many recent reports, Zeus has been considered to be the most serious botnet threat, with more than 3.6 million infected computers only in the U.S. [FC09]. The authors aimed to unveil the underlying architecture of the Zeus crimeware toolkit and enable its mitigation. A tool was also proposed to allow the extraction of the configuration information from the binary bot executables. Authors were also able to extract the encryption key that is used to encrypt the communications between the bots and

the C%C servers, which allows a direct interaction with the mentioned servers.

In the same year, a DNS-based detection approach was proposed to detect *botnet* collusion by analyzing anomalies in the degree distribution of visited domains [BSS10]. The work aims to detect C&C traffic that, as the authors state, is crucial for the *botnets* operation. The authors perform detection by observing the DNS traffic of a group of potentially infected computers and counting the number of computers that visit the same domain. Domains with a high number of visits can indicate a C&C domain and, using thresholds, legitimate domains are distinguished from the ones hosting C&C servers.

Using darknets

In an attempt to describe and understand the behaviors and the life-cycles of *botnets*, [ARZMT06] proposes a distributed measurement platform. Measurements were carried out during more than three months and more than 190 IRC *botnets* were tracked. The used data collection architecture is depicted in figure 2.6, and consists of three different logical phases. The first comprises collecting as many *bot* binaries as possible, while the second consists in analyzing the collected binaries using gray-box testing to extract the features of suspicious binaries. This is achieved by a two-phase procedure that includes (i) deriving a network signature of the analyzed binary and (ii) extracting the IRC-specific features. Finally, the third phase consists of tracking *botnets* using IRC and DNS trackers. The results achieved allowed authors to conclude that *botnets* are a major contributor to unwanted traffic in the Internet and that the scanning traffic generated by *botnets* differs from the traffic generated by malware (*worms*). This is most likely due to the human intervention that launches the scanning traffic in *botnets*.

In a recent work [SG10], the use of automated and self-adapting systems based on machine learning techniques was proposed. The authors analyzed information collected at three levels:

1. single packet level;
2. network access level;
3. TCP conversation level.

At the first level, the packets headers were analyzed in order to identify patterns, such as unusual combinations of flags and TCP options, that could indicate that they are malicious or spam. At the second level, the authors analyzed the access patterns of *bots* to a darknet, as well as the patterns of communications between bots and between bots and the command center. Such analysis is performed without looking into the contents of the packets. Finally, at the third level, the authors try to distinguish between legitimate and illegitimate TCP conversations. One example are the SMTP connections sending regular e-mails and the ones

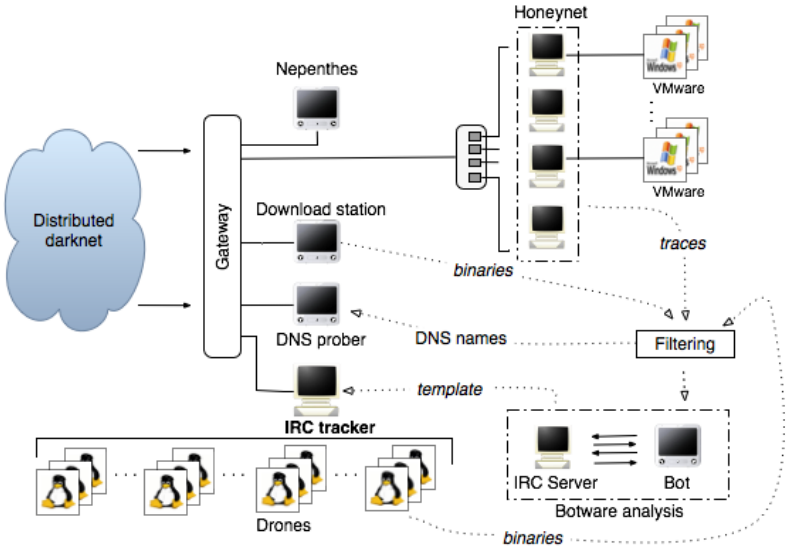

Figure 2.6: Overall data collection architecture proposed in [ARZMT06].

sending spam or *malware*. However, in this analysis, an inspection of the contents of the packets is performed and consequently, it must be restricted to unencrypted traffic.

The results obtained for the first level of analysis presented a very high accuracy on the identification of three of the eight considered spambots. A system to distinguish between static and dynamic IP addresses was also built, with an estimated accuracy of 99.15%. This system allowed authors to determine, as expected, that most *bots* IP addresses are dynamic. For the second level of analysis, the authors determined that different spambots presented similar access patterns, which indicates that these spambots could be controlled by the same operators. In the third level, the authors determined that although different spambots present similar access patterns, they generate very different types of spam and most of the spam data is sent in a single packet.

Reference [LWLS06] suggested a machine learning based approach for *botnets* detection using some general network-level traffic features of chat-like protocols such as IRC; in [BS06] authors combined IRC statistics and TCP work weight for the detection of IRC-based *botnets*; BotSniffer [GZL08] is an anomaly-based *botnet* C&C detection system that also used horizontal correlation, although it was mainly used for detecting centralized C&C activities (e.g., IRC and HTTP); reference [GPY+07] described BotHunter, a passive *bot* detection system that used dialog correlation to associate IDS events to a user-defined *bot* infection dialog model. Finally, the use of payload signatures and decision tree models was studied in [LTRG09].

2.5 User Profiling in Network Management

Profiling is a very useful tool for a variety of network management tasks, personalization and security. Indeed, in the recent years, *personalization* of web-contents and services became very important issues due to the emergence of many web-services such as e-commerce business models. In fact, it is argued that the ability of efficiently addressing the needs of a specific users can be economically more profitable than over more traditional segmentation methods. The main key technical issue is the development of accurate user profiles. There are several definitions for what a user profile is [GA05], but a common definition states that an user profile consists of a description of the user interests, behaviors and preferences [IALS11]. Therefore, it consists of a collection of user-related data that is adequate for the system [AT99]. User profiling can then be seen as the process of gathering, organizing and interpreting the user profile information. However, these definitions vary according to the classification objective. In our work, we define a profile as the set of Internet applications and services each user runs and interacts with. This definition is more appropriate for network management purposes since, as previously explained, by monitoring the users' traffic, mapping it into the originating Internet applications and building accurate user profiles, many network management tasks can be greatly improved.

A very important issue in user profiling is the set of rules used for building such profiles. In [AT99], the authors present a method for validating the set of rules used for building user profiles. The authors proposed a process for building user profiles that uses several data mining algorithms for discovering association and classification rules. Despite achieving efficient rules for building profiles, the authors always require human validation, although many approaches could be deployed for efficiently validating such rules.

Many works, like for example [IALS11], have addressed the issue of building accurate user-profiles that are able to describe the most important features. However, the set of features and, consequently, the definition of what is an user profile vary according to the objective of the classification. A pragmatical approach can consist in determining the domain name associated with the host/server that is being contacted. Subsequently, a simple association between the obtained domain and the services it runs can be performed [TRKN10, TRKN08]. In these works, the authors state that all information needed to profile any Internet endpoint is available around us - in the Internet. Therefore, in order to build an accurate profile authors simply have to query the most used search engine (Google) and divide the querying results into several tags describing the requested services. The obtained results proved that the approach is suitable for the proposed purpose, enabling even more accurate results than some of the state-of-the-art tools. Our work differs from the presented ones in the fact that a user profile is now defined as the set of web-based applications that are being used, that is, the focus is placed on applications that allow users to share on-line information and contents.

In [MSS+06] the authors built end-host profiles with the purpose of defending against

worm attacks. A profile is defined as the community of hosts an end-system normally interacts with, which is defined as the Community of Interest (COI). The authors exploit some properties of enterprise networks such as well-known topologies, knowledge of all end-hosts allowed and the control of all configurations in all routers and switches of the network. Training data was collected over several weeks to obtain a "normal communications pattern", which should restrict the ability of worms exploiting vulnerabilities on the network. The authors then build profiles that tolerate some deviation to normal profile in order to cope with changes on the network and on the user profiles. Finally, some known behavior of worms are used for assuring that the training data is free of attacks. As results, the authors discuss that rules should differentiate traffic running on fixed ports numbers from the traffic using random ports defined on-the-fly. Using the created profiles and rules, the authors were able to prevent attacks to the monitored networks. However, the accuracy of this approach depends on the profiles used and it is assumed that the patterns of an attack always deviate from normal communication profiles.

In [jKLLK10] a novel approach for building user profiles of concept networks for personalized search is proposed. The authors define and model a user profile as a networked structure of concepts, which are defined with the formal concept analysis that allows the use of Formal Concept Analysis (FCA) theory. A concept contains a user's query intention and reflects the user's preferences. Whenever a new query is issued, a session interest concept is generated and new concepts are then merged in the current concept network, *i.e* a user profile. Similarities between new concepts and the existing ones are also computed and a reference concept hierarchy is used for this purpose. The obtained results show that the proposed approach is able to improve the accuracy of search results in terms of personal preference.

Many network management tools are currently available, since this is also an active research field. One example is the Open Network Management System (Open NMS), an open-source platform that performs many network management tasks [NMS11], such as event and notification management, service assurance and performance measurement. Its scalability allows to monitor thousands of devices in a single network. However, the monitoring of the hosts service is based on the ports that the administrator associates with the different network applications, which constitutes a shortcoming of this platform. In [DP10] the growing complexity of the Internet and the increasing number of users and services are discussed for attaining flexible and scalable management solutions. Some important guidelines and research directions are provided to cope with the increasing complexity of the Internet and its applications.

2.6 Conclusions

In this section, the most relevant work in the different areas addressed by our classification methodologies has been presented. Such fields include traffic classification, intrusion and attacks detection and the detection of *botnets*. The issues associated to the different mentioned works have been discussed in an attempt to introduce some of the most important motivations that lead us to fulfill this work.

Chapter 3

Background

3.1 Introduction

In this chapter, several important background concepts are presented. We start by presenting the main Internet applications, the different mechanisms that generate and shape the traffic of each application and its main characteristics. Subsequently, the definition of *data-stream* is presented, together with its importance in the analysis of the different traffic dynamics. Having in mind that these dynamics are spread through the different sessions that are established with remote hosts and servers, the *data-stream* concept should be sufficiently comprehensive to incorporate them. The captured traffic and the studied Internet applications are then presented, together with an explanation of the capturing procedure.

A discussion on Fourier Transforms (FTs) and Wavelet Transforms (WTs), as well as a discussion on the advantages and limitations associated to each decomposition approach, is also provided. Then, the Multi-Scale Traffic Analysis methodology is proposed, together with a description of which dynamics are intended to be analyzed by this approach. The chapter will then proceed with some important preliminary definitions that will be intensively used in subsequent chapters. Finally, the classification metrics used to evaluate the accuracy of the proposed classification approaches are also presented.

3.2 Internet Traffic, Internet Applications and their Dynamics

The Internet is a global network of interconnected networks comprising billions of users worldwide. As already mentioned in chapter 1, in recent years this Network has grown in size, complexity and importance. In addition, the recent and stunning increase on the services and applications available implied implementing novel communication paradigms and transporting different types of data, such as files, voice, video and many more. The well known Internet Protocol (IP) is used to connect and transport this data between all the connected computers and it can be seen as the universal language of the Internet, understood

by every computer and connected device, thus enabling internetworking. Despite being able to forward packets throughout the Internet, it does not guarantee their successful and timely delivery to its intended recipients. These tasks have to be assured by other transport-layer specific protocols, such as TCP, which ensure a reliable and ordered delivery of the requested packets. TCP exploits the ability of the IP protocol to forward and understand the headers of the packets and implements appropriate control channels that enable the detection of dropped packets and packets delivered out of order. On top of this, applications are able to transmit data over the Internet by requesting it to the transport layer and letting it handle all the transport sessions establishments and IP packets transmissions. The encapsulation of the different layers and their corresponding network protocols is a very important concept in networking, which leads to the creation of different frequency components on the Internet traffic [FGW98].

A very important aspect of Internet applications is that each application requires different user interactions, according to the implemented service, and generate different interactions with remote hosts and servers that lead to the creation of different traffic dynamics. For instance, web-browsing applications require frequent user interactions that originate different traffic peaks corresponding to user requests and to the subsequent download of web pages. In addition, the number of simultaneously contacted HTTP servers is typically low. These applications are also more tolerant to delay and jitter and do not perform a large bandwidth consumption. On the other hand, video applications do not require so frequent user interactions and present a constant and considerable bandwidth consumption due to the transmission of videos. These applications are sensitive to delays and jitter, since they affect the quality of the video reception and perception. The number of contacted servers is typically very low, leading to the creation of a single Internet session with the server that hosts the video. Applications involving the download of large files over P2P networks generate even more different traffic patterns, which can be characterized by a large and varying bandwidth consumption and, like web-browsing, are also tolerant to delays that may occur on the links. The number of simultaneously contacted hosts is high, since such applications enable the simultaneous download of different file chunks from different hosts in order to speed up the file transfer.

On the other hand, several events and mechanisms shape Internet traffic and create its different frequency components. For instance, an Internet user performing a request on a web-application, such as web-browser, creates a set of Internet sessions that, in turn, create a set of Internet packets that are transmitted over the physical connecting medium. These events create several frequency components in different frequency spectrum regions. This concept is illustrated in figure 3.1, which shows three different frequency spectrum regions, together with their corresponding events. Low frequency components account for human events that, in the Internet world, are associated with human/user behaviors and actions. Between the low and high frequency regions, we have created a mid-range frequency region that accounts

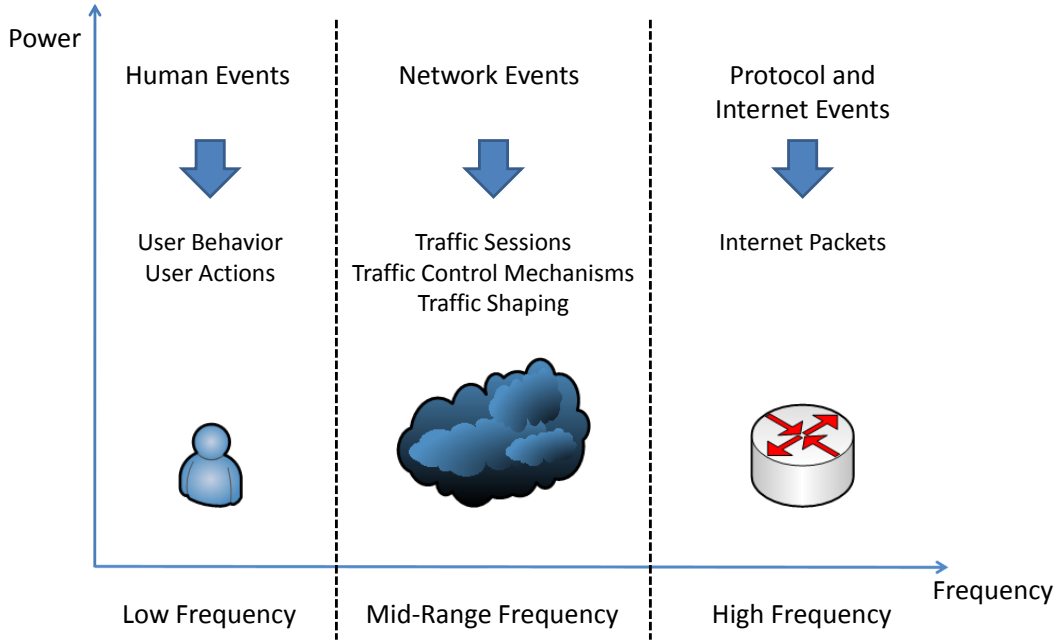

Figure 3.1: Frequency regions mapping into network and users mechanisms.

for network events such as the creation of traffic sessions and the corresponding traffic control mechanisms. Other used control mechanisms, such as traffic shaping, are also covered by this region. Finally, in the high-frequency spectrum region, protocol and Internet events, such as packets arrivals, are accounted for. Internet applications presenting these components are the ones that generate a considerable amount of traffic with a high number of received packets. All these frequency components are spread over the different simultaneous interactions that are generated by an Internet application with the various remote clients and servers. The analysis of such components is critical for achieving an efficient differentiation between the dynamics generated by the different Internet applications. In the following sub-section, a novel traffic definition will be presented to address this issue.

3.2.1 Data-Streams Definition

As previously mentioned, an Internet application generates several and simultaneous interactions with remote hosts and/or remote servers that lead to the creation of very different traffic dynamics. Traditionally, Internet traffic is grouped in flows according to the classic *five-tuple* definition:

- source and destination IP addresses;
- source and destination port numbers;
- transport-layer protocol.

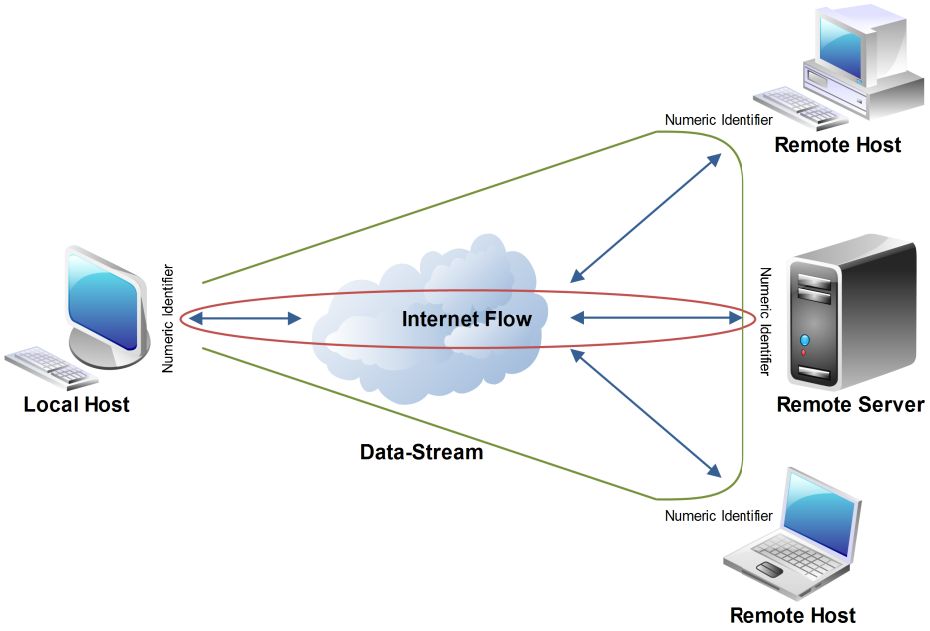

Figure 3.2: Traffic generated by an Internet application: *data-streams* vs Internet flows.

However, such flows only correspond to one of the many interactions generated by an Internet application. In order to be able to analyze and classify such interactions and their different frequency components, the restrictive classical definition of Internet flow was replaced by the definition of *data-stream*. This consists of all traffic (in the upload or download directions) of a local IP address that is univocally identified by a numeric identifier that can be defined as:

1. a specific TCP/UDP (local or remote) port number - for unencrypted traffic;
2. the Security Parameters Index (SPI) in ESP headers, in the case of IPsec tunnels, or any other specific identifier of IP-level encrypted tunnel technology - for encrypted traffic.

Therefore, *data-streams* are uniquely identified by a *2-tuple* (IP address, unique identifier). We use this definition since we strongly believe that the analysis of the different simultaneous interactions generated by an Internet application as a whole provides a deeper insight into how applications behave. The analysis of such frequency components can play an important role in traffic discrimination. The presented concept is illustrated in figure 3.2, which shows a comparison between *data-streams* and the classical Internet flows.

In addition, we can define *known data-streams* as *streams* that are analyzed *a priori* to determine its originating application(s) and *unknown data-streams* as traffic *streams* that are created by an unknown application that will be used to assess the accuracy of the proposed classification methodologies.

3.2.2 Traffic Traces

This sub-section presents the real traffic traces that were captured to evaluate the accuracy of the different proposed classification methodologies. The traffic of the several studied Internet applications was passively collected at the communications network of University of Aveiro and was measured between September 2008 and September 2010. The captured traffic is composed by unencrypted IP/TCP and IP/UDP packets, while TCPDump, publicly available at [TCP11], was used to capture the full header and the first 10 payload bytes. In order to verify the ability of the different proposed classification and identification methodologies, traffic was divided in two categories:

1. Licit Applications: used to assess the ability of identifying and classifying legitimate Internet traffic;

2. Illicit Applications: used to assess the ability of identifying and classifying traffic with illicit patterns and low-impact and stealth anomalies.

    - These consist of intrusion attempts or traffic corresponding to information theft that our classification methodologies must correctly identify.

Let us first present the traffic of the studied legitimate applications, how it was captured, the protocols that were used and the traffic patterns that were obtained.

Licit Applications

In order to evaluate the ability of the proposed classification methodologies to provide an accurate identification of legitimate Internet applications, we divided them into three main categories:

- Web-Browsing - browsing though websites, reading available information and pressing the available links to request other web pages;

- Video Streaming - watching a video from a Television channel website;

- Large Files Download - download of files through P2P networks and protocols.

We captured and analyzed traffic belonging to all these licit Internet applications. Let us present the traffic that was generated and captured for each application, as well as the clients that were used to generate it. All traffic was sampled at a rate of 100 ms and the time series that were extracted are the number of captured bytes and packets per sampling interval, together with their arrival instants.

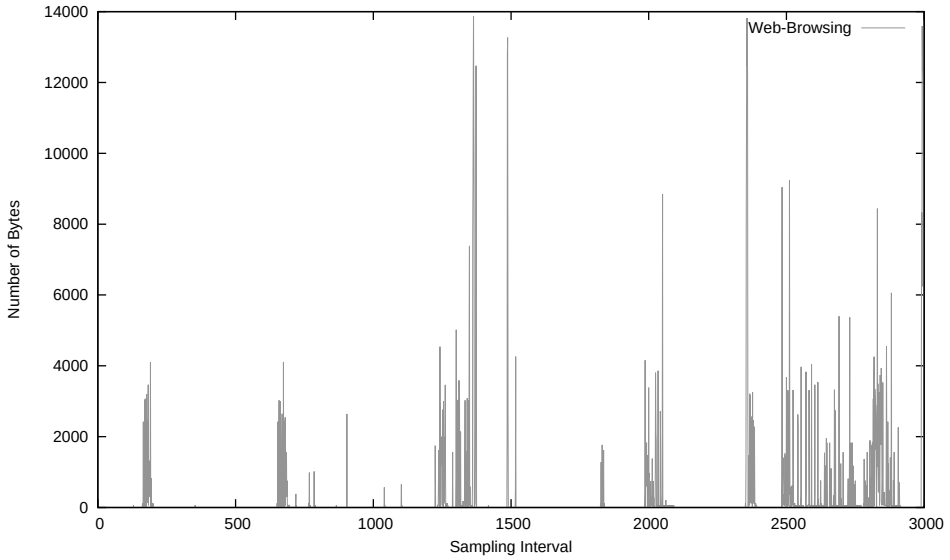

Figure 3.3: Sample Web-Browsing traffic for the upload and download directions.

Web-Browsing Web-browsing traffic was captured by browsing through the most important Portuguese newspaper web-sites, reading the available news. Traffic was captured for both the upload and download directions and the number of bytes per sampling interval is shown in figure 3.3.

Several non-periodic and very short duration peaks can be observed, corresponding to the user requests and the subsequent download of the requested pages. The non-periodicity of these peaks is related to the user profile and usage of the on-line news services, which is a characteristic of Web-browsing traffic.

Video Streaming Video Streaming traffic was generated by using the streaming services offered by some important Portuguese television channels that are available on their web-sites. Therefore, the Streaming service ran on top of the HTTP protocol and the main used channel was SIC [SIC11]. Such solution is in fact the cheapest and simplest manner of streaming video contents from a website. The captured traffic (number of bytes per sampling interval) is shown in figure 3.4, where we can see that the profile generated by this application consists in a constant bandwidth consumption with small variability, which is due to the transmission of the requested video. In addition, some periodic and very short duration peaks, with significant absolute values, can also be observed, corresponding to the synchronization between the client and the Streaming server.

Large Files Download One of the main uses of the Internet is the download of large files. Such downloads are usually performed using P2P networks and clients. In our work, we used the most deployed P2P client/protocol: BitTorrent [BTS09]. The captured traffic is shown

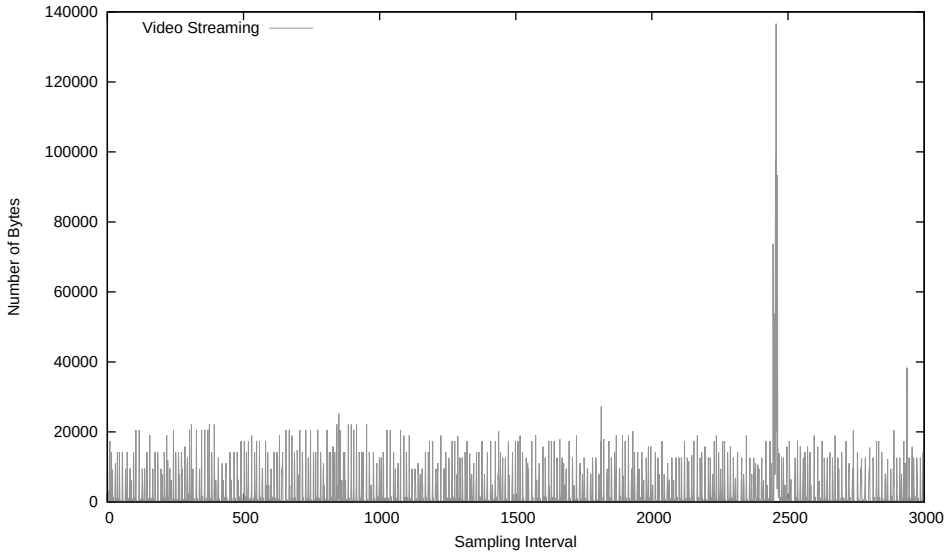

Figure 3.4: Sample Video-Streaming traffic for the upload and download directions.

in figure 3.5 and presents a high and constant bandwidth consumption, with a noticeable variability around the average bandwidth.

Illicit Applications

In order to assess the accuracy of detecting illicit applications and traffic presenting illicit patterns, some widely deployed low-impact and stealth attacks were emulated at our lab. Specifically, host scans and information theft were deployed in order to replicate the behavior of compromised hosts. The following sub-sections will present the emulated attacks.

NMap/Host Scans The first emulated attack consisted in host scans replicating the behavior of a compromised host instructed to scan its neighbor hosts in order to determine their open ports and, consequently, the services they run. Subsequently, the vulnerabilities associated to those services can be exploited for gaining access/control to the attacked host [NI07]. The well-known NMap application [NMa11, Lyo09] was used to scan hosts in our research lab. A discrete profile was used in order to bypass possible protection and detection mechanisms, such as IDSes and proxies: this profile consisted of a sequential port scan with one second of interval between (SYN) probes and a waiting time of 15 seconds. The traffic generated by these scans is shown in figure 3.6.

Snapshots/Information Theft Snapshots/Information theft is one of the most common attacks in compromised hosts and can be achieved by capturing snapshots, which are pictures of the current screen contents showing what is being displayed, and sending them to the

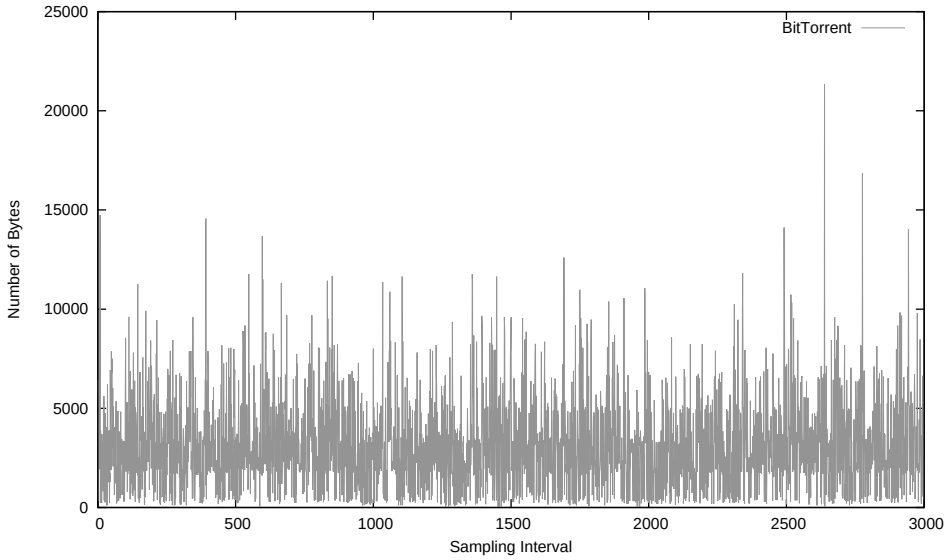

Figure 3.5: Sample BitTorrent traffic for the upload and download directions.

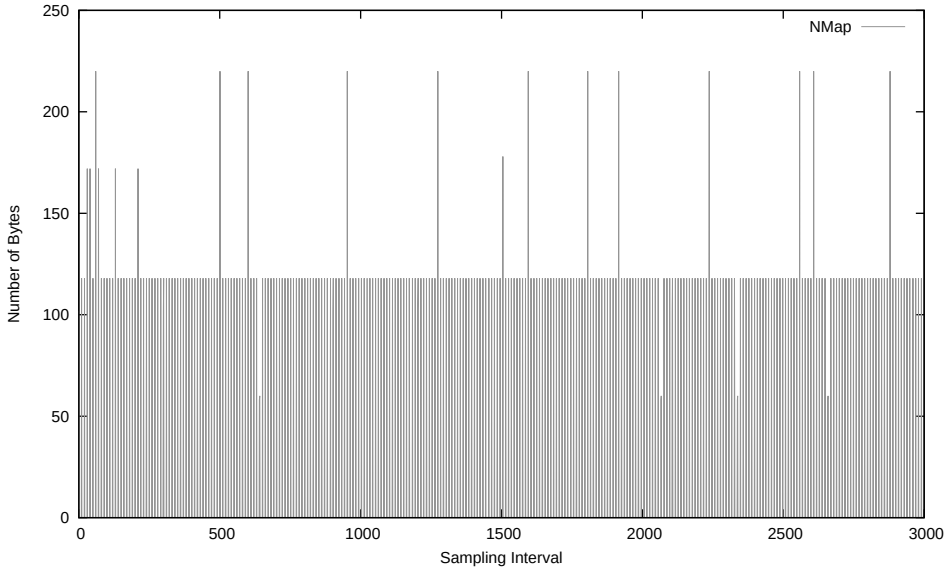

Figure 3.6: Sample NMap traffic for the upload and download directions.

attacker. Such snapshots can be captured when a trigger, such as a key word appearing in a window of a browser or on the URL of a page, occurs [NI07]. To emulate such type of attacks, we have captured small pictures (335x180 pixels, 120KBytes) of a host in our research lab and uploaded it to its *bot-master* via FTP. We assumed that the user was browsing the Web and performed requests with an exponentially distributed interval with average equal to 120 seconds [ZAN99]. Therefore, the uploads of the captured snapshot were performed

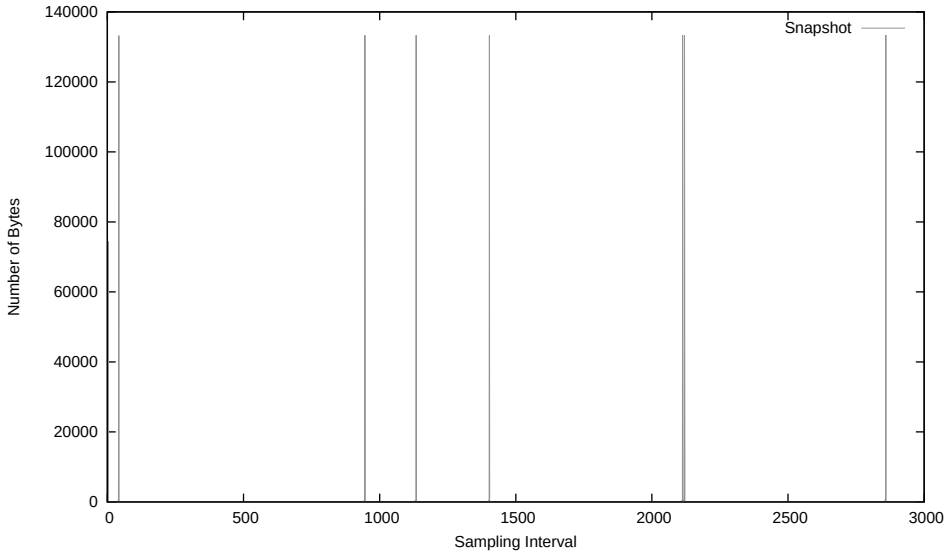

Figure 3.7: Sample Snapshot traffic for the upload and download directions.

according to this exponential distribution. The captured traffic, in the upload and download directions, is shown in figure 3.7. Some non-periodic peaks can be seen, corresponding to the establishment of the FTP connection with the remote host/server and the subsequent upload of the captured file.

3.3 Traffic Scaling Analysis

In this section, the most relevant methodologies for the analysis of the frequency components of signals and time-series are discussed. We start by presenting Fourier Transforms (FTs) and describing their mathematical formulation. The advantages and drawbacks associated to this approach are also discussed, followed by a description of WT, which is a signal analysis methodology that is able to address some of the issues associated to FTs. Finally, we will present our multi-scale traffic analysis approach, explaining how wavelets are used in the approach.

3.3.1 Fourier Transform

The FT is the most used technique for analyzing the frequency spectrum of a stochastic process, by decomposing it into complex exponential functions having different frequencies [Mor96].

Let us define $L^2(\mathbb{R})$ as the set of square and integrable functions, *i.e*, the set of real functions $x(t)$ satisfying:

$$\int_{-\infty}^{+\infty} |x^2(t)|dt < \infty \tag{3.1}$$

with the inner product defined as:

$$\langle x, y \rangle = \int_{-\infty}^{+\infty} x(t)y^*(t)dt \tag{3.2}$$

and norm

$$\|x\| = \langle x, x \rangle^{1/2} \tag{3.3}$$

Subsequently, the Fourier transform of a function $x(t) \in L^2(\mathbb{R})$ can be defined as:

$$X(w) = \int_{-\infty}^{+\infty} x(t)e^{-iwt} \tag{3.4}$$

where w denotes the frequency of the analyzing sinusoid. Since the support of the sinusoid is not localized, FTs have a poor time resolution and are only suitable for the analysis of stationary signals, *i.e*, signals presenting the same frequency component in the whole range of analysis. Consequently, FTs are unable to provide time-frequency representation, where the different frequency components of a non-stationary process are depicted together with the time-intervals where they occur. Therefore, time-varying signals or signals with transient and/or sudden changes require other analysis tools [Mor96].

3.3.2 Wavelets

As mentioned in sub-section 3.3.1, FTs require that the analyzed signal is stationary, that is, the frequency components of the analyzed data do not change over time. In many cases, such restriction is respected by the analyzed data and an accurate decomposition can be achieved. However, this is not the case with Internet traffic, which is known to be non-stationary since it presents different frequency components at different time intervals. By assuming non-stationarity [TC98], WTs are able to provide a time-frequency representation of a signal and are widely applied in many different areas such as signal processing, image analysis and compression, turbulence analysis and analysis of stocks market exchange rates. In fact, this is a powerful technique for understanding the complexity of real world processes.

Wavelets are mathematical functions that are used to divide a given signal into its different frequency components. They were introduced in 1980 by geophysicist J. Morlet to perform signal decomposition and approximation, and consist of a short duration wave-like oscillation with a limited amplitude, occurring during a short period of time that gives it a good time and frequency resolution. Wavelets enable the analysis of each one of the signal components in an appropriate scale and present several advantages over other signal analysis techniques,

such as Fourier Transforms. As already mentioned, FTs are more suitable to analyze periodic data, while WTs are more adequate to analyze functions with discontinuities and peaks. Since wavelets present a compact support, they present a very good time resolution and can, consequently, provide information concerning both time and frequency, while FTs only provide frequency information. In addition, the infinite set of basis functions for wavelets is another important advantage over Fourier Transforms, which use a finite set of basis functions (*sines* and *cosines*).

A wavelet $\psi(t)$ can be defined as a pass-band function oscillating at a central frequency f_0, satisfying the admissibility condition [YY94, Dau92]:

$$0 < C_\Psi = 2\pi \int_{-\infty}^{+\infty} \frac{|\Psi(w)|^2}{|w|} dw < \infty \tag{3.5}$$

where C_Ψ is the *admissibility constant* and $\Psi(w)$ is the FT of ψ. To achieve this condition, it is sufficient that the mean of the function vanishes, that is:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \tag{3.6}$$

which implies that wavelets must have a band-pass like spectrum and a wave-like form. Such properties enable an effective localization in both time and frequency, as opposite to FTs. The wavelet ψ is designated as the *mother wavelet* and one example is shown in figure 3.8.

The following sections will present the two different types of WTs: the Continuous and Discrete Wavelet Transforms. The characteristics of each type of transform will be discussed, as well as its most appropriate usage scenarios.

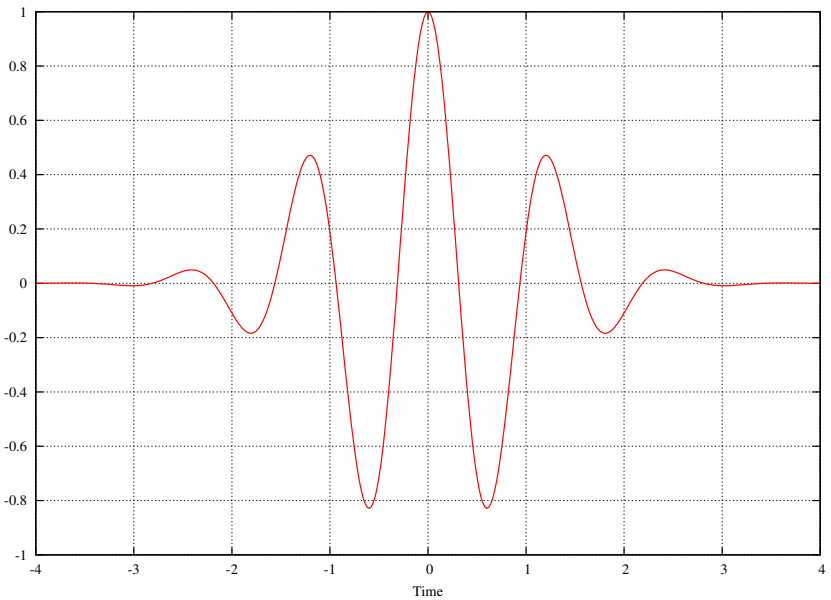

Figure 3.8: A typical wavelet.

Continuous Wavelet Transform (CWT)

The use of a wavelet decomposition based on a CWT allows the analysis of any process in both time and frequency domains. By scaling and translating the mother wavelet ψ, a set of functions $\psi_{\tau,s}$, the *wavelet daughters*, can be generated:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{|s|}}\psi(\frac{t-\tau}{s}), \quad s, \tau \in \mathbb{R}, s \neq 0 \tag{3.7}$$

where τ and s are the translation and scale parameters, respectively. The first parameter is used for shifting the mother wavelet in time, while the second parameter controls the width of the window analysis and, consequently, the frequency that is being analyzed. Since this is a continuous transform, both τ and s must be incremented continuously and the transform has to be integrated over all time, which makes this transform a heavy computational task. By varying these parameters, a multi-scale analysis of the entire captured process can be performed, providing a description of the different frequency components present in the decomposed process together with the time-intervals where each one of those components is located. Given a time series $x(t) \in L^2(\mathbb{R})$, its CWT $C_x^{\psi}(\tau, s)$ can be defined as [SSB03]:

$$C_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{+\infty}^{-\infty} x(t)\psi^*(\frac{t-\tau}{s})dt, s, \tau \in \mathbb{R} \tag{3.8}$$

where $*$ denotes the complex conjugation of the base wavelet function $\psi(t) \in L^2(\mathbb{R})$ and $\frac{1}{\sqrt{|s|}}$ is used as an energy preservation factor.

By analyzing the original time series in the whole range of decomposition scales, CWTs are able to provide a representation of that series in the time and frequency domains. A Wavelet Scalogram can be defined as the normalized energy $\hat{E}_x(\tau, s)$ over all possible translations (set $\mathbf{T}$) in all analyzed scales (set $\mathbf{S}$), and is computed as:

$$\hat{E}_x(\tau, s) = 100 \frac{\left|\Psi_x^{\psi}(\tau, s)\right|^2}{\sum_{\tau' \in \mathbf{T}} \sum_{s' \in \mathbf{S}} \left|\Psi_x^{\psi}(\tau', s')\right|^2} \tag{3.9}$$

The volume bounded by the surface of the scalogram is the mean square value of the process. The analysis of these scalograms enables the discovery of the different frequency components, for each scale (frequency) of analysis. For instance, the existence of a peak in the scalogram at a low frequency indicates the existence of a low-frequency component in the analyzed time-series, while a peak in the scalogram at a high-frequency corresponds to an existing high-frequency component. In addition, assuming that the process $x(t)$ is stationary over time, several statistical information, such as the standard deviation, can be obtained:

$$\sigma_{x,s} = \sqrt{\frac{1}{|\mathbf{T}|}\sum_{\tau\in\mathbf{T}}(\hat{E}_x(\tau,s) - \mu_{x,s})}, \forall s \in \mathbf{S} \tag{3.10}$$

where $\mu_{x,s} = \frac{1}{|\mathbf{T}|}\sum_{\tau\in\mathbf{T}}\hat{E}_x(\tau,s)$, and $|\mathbf{T}|$ denotes the cardinality of set $\mathbf{T}$.

Since it is a computationally heavy task, this transform is more suitable for off-line procedures. In our work, this approach was used to perform user profiling, as presented in chapter 7, since such profiling is a task that is performed off-line.

Discrete Wavelet Transform (DWT)

DWTs can be also used to represent functions and signals both in their time and frequency components. Another advantage of DWTs is the fact that they are computationally less complex, since its complexity is *O(N)* while the complexity of the Fast Fourier Transform is *O(Nlog(N))*. This makes such transforms suitable for real-time tasks, which in this thesis comprise traffic decomposition, analysis and classification. Consequently, DWTs will be intensively used in our traffic classification approaches, proposed in chapters 5 and 6. These approaches are suitable for deployment in network traffic classifier modules, as already presented in section 1.3.

By performing a scaling change, which may consist of an expansion or a compression, and a temporal shift on the mother wavelet, we obtain $\psi_{j,k} = 2^{-j/2}\psi(2^{-j}t - k)$, that is the oscillating central frequency moves to $2^{-j}f_0$ and the origin of the temporal reference to $2^j k$. Note that j represents the temporal scale, k represents the k^{th} coefficient corresponding to scale j, with j_0 being the largest time scale. DWTs also use a low-pass function, $\phi(t)$, known as scaling function, that can be scaled and temporarily shifted in a similar way to function $\psi(t)$. Therefore, a signal $x(t)$ can be built as a sum of the scaling and wavelet functions:

$$x(t) = \sum_k c_x(j_0,k)\phi_{j_0,k}(t) + \sum_{j=j_0}^{\infty}\sum_k d_x(j,k)\psi_{j,k}(t) \tag{3.11}$$

where $\phi_{j_0,k}(t)$ and $\psi_{j,k}(t)$ are, respectively, a generic scaling function and a generic wavelet function. $c_x(j_0,k)$ are the scaling coefficients and $d_x(j,k)$ are the wavelet coefficients. The logarithm of the wavelet coefficients, for the moment of order q, of the wavelet coefficients can be defined as:

$$y_{q,j} = \log_2\left(\frac{1}{K}\sum_{k=1}^{K}|d_x(j,k)|^q\right), q \in \mathbb{R} \tag{3.12}$$

and will be hereafter generically designated as multi-scale estimators, where K is the number of coefficients to be analyzed at time scale j. The scaling behavior of any stochastic process can be studied by an analysis of the Log-scale Diagram (LD), which is a log-log plot of

the estimators $y_{q,j}$ of the wavelet details at each scale, against scale, completed with the Confidence Intervals (CIs) about the estimates at each scale [AFTV00].

3.3.3 Multi-Scale Traffic Analysis

In our work, we will use wavelets to analyze and decompose network traffic at several scales, *i.e* different aggregation levels, in order to evaluate and correlate the different characterizing frequency spectrum components and the corresponding underlying network mechanisms. As explained in section 3.2, Internet traffic is generated by low-frequency events such as user requests and controlled by components present in the mid-range frequency spectrum, which account for the creation of Internet sessions and the different existing traffic control mechanisms. All these events and components create high-frequency events that correspond to the arrival of Internet packets. For instance, when a user performs a request using an Internet application, such as clicking on a link in a web site or requesting an on-line video, several processes are created by the operating system. Each one of these processes creates a set of Internet sessions, each generating a traffic flow. At the network layer, each one of these connections will transmit and receive the requested data in several packets. This is shown in figure 3.9, which illustrates how the mechanisms present in the different scales are related, how they shape Internet traffic and how they can be analyzed. By analyzing traffic generated by an Internet application, shown in the left side of the figure, and zooming into the observed dynamics we are able to infer all the mechanisms present at the different scales of analysis and assess their influence in the global dynamics of the traffic. Components such as the time intervals between user requests (represented by Δ_1), their starting instants (represented by Δ_{2x}) and predominance can be evaluated by performing a change on the scale of analysis, which corresponds to perform a *"zoom in"* in the analyzed traffic. Finally, the components created by Internet packets, their arrival instants (represented as Δ_{3x}) can also be evaluated by performing another change on the scale of analysis, which corresponds to another *"zoom in"* in the analyzed traffic. The main concept of our approaches and analysis consists in evaluating the presence of each one of the mentioned mechanisms, which can be done by using the appropriate scale of aggregation, or frequency scale. In this manner, we aim to obtain characteristic spectral signatures describing the several frequency components, for each studied application, which will enable an accurate traffic discrimination.

The underlying concept of our approach consists of analyzing the several interactions created by an Internet application. These may consist of several and simultaneous sessions with different remote hosts and servers. For unencrypted traffic, the traffic of the different applications can be monitored separately, while for unencrypted traffic this may not be possible as illustrated in Figure 3.10. Therefore, we have created the definition of *data-stream*, presented in section 3.2.1, which consists of all traffic that is sent and received by an Internet application class and identified by a numeric identifier. Such identifier can include the (i)

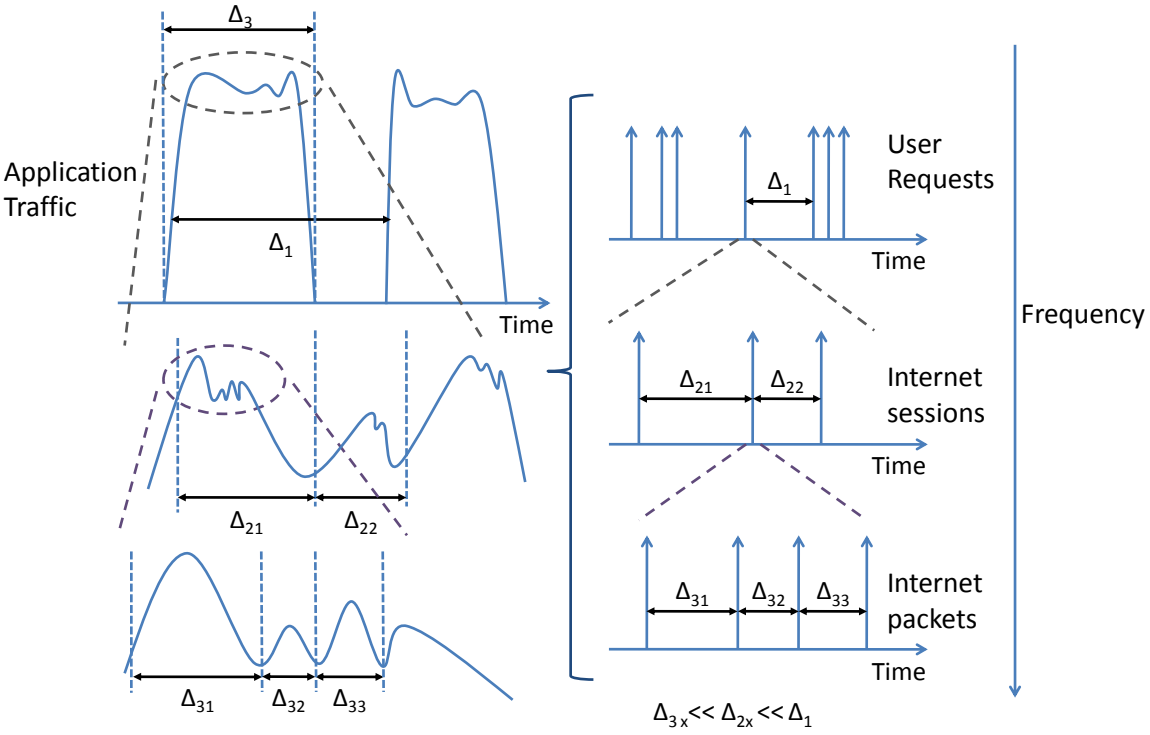

Figure 3.9: Multi-Scale Traffic Dynamics.

local/remote port (for unencrypted traffic) or (ii) any specific identifier of IP-level encrypted tunnel technology (for encrypted traffic). For the latter case, the samples that are analyzed in a *sampling-window* of a *data-stream* can consist of traffic generated by several applications running simultaneously on a client. However, if the traffic captured in a time interval is generated by only one application, then in the corresponding *sampling-window* the components of the *Multi-Scale Signature* of the corresponding application class can be observed and identified. The analysis of *data-streams* over sliding *sampling-windows* of a pre-defined length Δt, as illustrated in Figure 3.10, from where traffic samples are extracted enables a continuous monitoring of the traffic of each application, increasing the classification accuracy of our approach. This also enables the identification of stealth, low impact and distributed threats/anomalies.

This multi-scale analysis for the traffic classification approaches was enabled by using the tool available at [Dar08] that estimates the *q-th* order wavelet estimators, based on a DWT. This methodology examines the behavior of the *q-th* order moment estimators over a set of aggregation/decomposition scales thus enabling the analysis of all the frequency components present in the analyzed traffic. On the other, the multi-scale analysis for the user profiling approaches were enabled by implementing the CWT and decomposing the extracted traffic metrics using such implemented functions.

3.3.4 Some preliminary definitions

In this section, some important definitions that are common to all classification approaches will be presented. Let us begin by defining the two different types of *data-streams* that are used in this thesis. The first consist of *known data-streams*, that is, traffic identified via *deep-packet inspection*, whenever possible, or traffic generated in a controlled lab environment. These are used for labeling the classification clusters and for inferring the parameters of the different probabilistic distributions. The second type consists of *unknown data-streams* and are used to assess the accuracy of the proposed methodology.

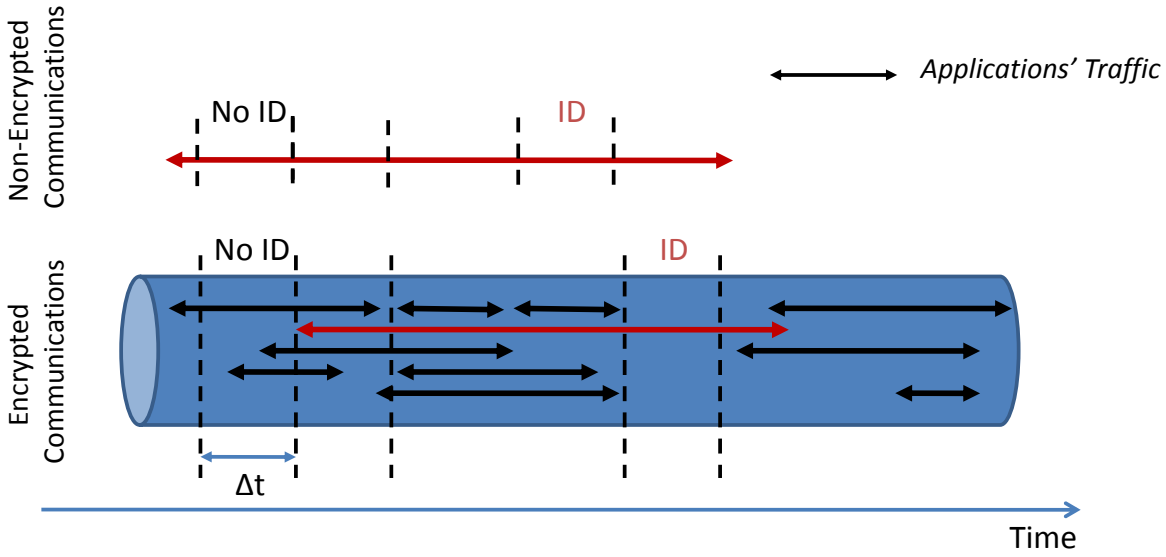

Figure 3.10: Traffic classification concept.

Let A represent the number of studied applications, M represent the number of *unknown data-streams* of each Internet application that will be classified, N correspond to the number of *known data-streams* of each Internet application, S represent the number of stochastic processes that are extracted from the *data-streams*, Q represent the number of different statistical moments that are considered and J the number of time-scales considered in each individual DWT analysis. In addition, let

$$\Gamma_z = (s, q, j) \in \mathbb{N}^3, s = 1, \ldots, S, q = 1, \ldots, Q, j = 1, \ldots, J \tag{3.13}$$

represent the $z - th$ element of the set

$$D = \{\Gamma_z, z = QJ(s-1) + J(q-1) + j\} \tag{3.14}$$

that indexes all the available stochastic processes, moments and time-scales. Moreover, let

$$E_{a,\Gamma_z} = \{e^i_{a,\Gamma_z}, i = 1, \ldots, N\} \tag{3.15}$$

represent the set of the estimators, as defined in (3.12), obtained from the element Γ_z that indexes a scale $j, j = 1, \ldots, J$, at the order $q, q = 1, \ldots, Q$, moment of the wavelet estimators obtained from a multi-scale analysis of a stochastic process $s, s = 1, \ldots, S$, of a *known data-stream* $i, i = 1 \ldots, N$, of an application a.

On the other hand, let us define

$$U_{\Gamma_z} = \{u^i_{\Gamma_z}, i = 1, \ldots, M\} \tag{3.16}$$

as the set of the estimators obtained for the element Γ_z that indexes a scale $j, j = 1, \ldots, J$, at the order $q, q = 1, \ldots, Q$, moment as defined in (3.12), obtained from a multi-scale analysis of a stochastic process $s, s = 1, \ldots, S$, of an *unknown data-stream* $i, i = 1 \ldots, M$.

3.4 Classification Metrics

An important criterion when evaluating the accuracy of traffic classification approaches is their accuracy [NA08]. A set of metrics have been proposed, the most common being:

- *False Positives (FPs)*;
- *False Negatives (FNs)*;
- *True Positives (TPs)*;
- *True Negatives (TNs)*.

Assuming a particular traffic class X and assuming that the classifier has two outputs (member belonging or not to class X), these metrics can be defined as follows:

- *False Positives (FPs)*: percentage of members of other classes incorrectly assigned to class X;
- *False Negatives (FNs)*: percentage of members of class X incorrectly assigned to other classes;
- *True Positives (TPs)*: percentage of members of class X correctly assigned to class X;
- *True Negatives (TNs)*: percentage of members of other classes correctly assigned to other classes;

The relations between the different classification metrics are shown in figure 3.11. For simplification purposes, the positive and negative results correspond to a member of a class being assigned, or not, to the correct class. These metrics will be used to evaluate the accuracy of our classification methodologies.

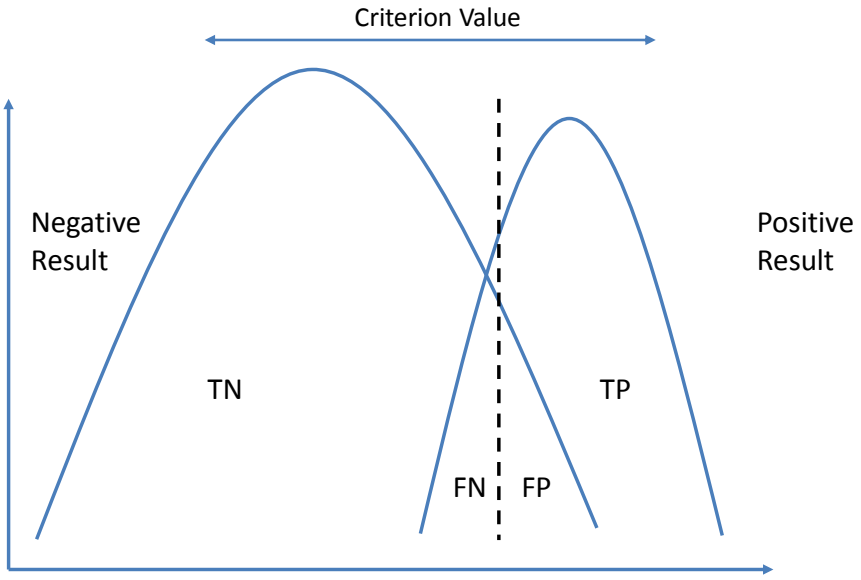

Figure 3.11: Relations between the several classification metrics.

3.5 Conclusions

Several background concepts, which are intensively in the following chapters, were presented in this section. We began by providing an explanation on the dynamics of the traffic generated by different Internet applications. Then, the definition of *data-stream* was presented, also explaining how traffic can be grouped into *data-streams* according to the classification restrictions. This chapter also presented the different legitimate Internet applications that were studied, as well as the emulated security threats. The captured traffic was also presented. Some notions on traffic scaling analysis were also presented, as well as a discussion on the most relevant existing methodologies for signal frequency analysis. The advantages and issues associated to each approach were also discussed, followed by a presentation of the different multi-scale traffic analysis approaches that will be proposed in this thesis. Some preliminary definitions were also presented, as well as the classification metrics that will be used throughout the thesis.

Chapter 4

Traffic Classification based on Clustering of the Multi-Scale Decomposition Estimators

4.1 Introduction

This chapter presents the first proposed classification methodology which is based on the usage of clustering algorithms for grouping the estimators that present the same behavior over the analyzed decomposition scales. By studying the components present in the first scales of the first and second order moment of analysis of the estimators of the *known data-streams* of each Internet application, we are able to build different groups for each Internet application to which the estimators of the *unknown data-streams* can be assigned to. In this manner, the corresponding *data-streams* can be classified. Groups will be built based on clustering algorithms, so the next paragraphs will present some important background on this issue. Subsequently, a presentation of the classification methodology and of the obtained results is provided. Finally, some conclusions are discussed.

4.2 Definitions

Clustering aims to partition a set of objects into groups, or clusters, in such a way that objects in the same group are similar, whereas objects in different clusters are distinct. The creation of clusters is based on the concept of proximity between objects and groups of objects [KR90]. There are two common approaches to cluster observations: the hierarchical and non-hierarchical, among which the partition methods are the most common.

Hierarchical clustering techniques proceed by either a successive series of merges (agglomerative hierarchical methods) or by successive divisions (divisive hierarchical methods). The

agglomerative methodologies start with as many clusters as objects and end with only one cluster, containing all objects. These are based on a measure of proximity between two objects and a criterion, relying on the distance between clusters, to decide which are the two closest clusters to be merged in each step of the agglomerative hierarchical procedure. Different approaches to measure the distance between clusters give rise to different hierarchical methods. A widely used method is the *Wards's method*, also known as the *incremental sum of squares method*, that uses the (squared) within-cluster and between-cluster distances to decide which clusters should be merged. Divisive methods work in the opposite direction.

Partitioning non-hierarchical clustering consists in dividing the data set into a predetermined number of non-overlapping clusters, so that each data object belongs to a cluster. One example of a partitioning clustering methodology is the *K-Means* algorithm [Mac67], which is also one of the simplest deploying a squared error criterion which acts as an *objective function*. It starts with a random initial partition and performs several reassignments of the analyzed patterns to the clusters based on similarity measures between the patterns and the clusters [JMF99]. The algorithm builds spherical clusters and attempts to find a user-chosen k number of clusters in the data set in such a way that they should be disjoint and represented by their centroid. Within each cluster, this algorithm maximizes the homogeneity through a minimization of the Mean Squared Error (MSE) which is computed as follows:

$$MSE = \sum_{i=1}^{k} \sum_{j=1}^{n} |x_j - c_i| \tag{4.1}$$

where c_i represents the centroid of the cluster i and x_j represents the *j-th* data point contained in the mentioned cluster.

The algorithm starts by randomly choosing the centroids of the K clusters. Subsequently, objects are assigned to the closest cluster and, then, the centroids of each cluster are iteratively re-computed and re-partitioned according to the new centers. This process continues until all members inside each cluster stabilize. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of points in $\mathbb{R}^d$. After randomly choosing the K centers $c_1, c_2, \ldots, c_K \in \mathbb{R}^d$, the data points are assigned to the corresponding cluster as follows [EAM06]:

1. For each $i, j \in \{1, \ldots, k\}$, set the cluster C_i as the set of points in X that are closer to c_i than to $c_j, \forall j \neq i$;

2. For each $i \in \{1, \ldots, k\}$, recompute the clusters centroids c_i as the center of all points, using the new membership in $C_i : c_i = \dfrac{1}{|C_i|} \displaystyle\sum_{x_j \in C_i} x_j$;

3. If a converge criterion is not met, repeat steps 1 and 2. Otherwise, return the created clusters.

Typical convergence criteria include no reassignment of any pattern from a cluster to any other or no decrease, after some iterations, on the MSE.

Another example of a clustering algorithm is DBSCAN (Density Based Spatial Clustering of Applications with Noise), which is a density-based algorithm. Density-based algorithms regard clusters as dense areas of objects that are separated by less dense areas. These clustering algorithms have an advantage over partition-based algorithms because they are not limited to finding spherical shaped clusters but can find clusters of arbitrary shapes. The DBSCAN algorithm is based on the concepts of density-reachability and density-connectivity, notions that are used to define what the DBSCAN algorithm considers as a cluster: a cluster is defined as the set of objects in a data set that are density-connected to a particular core object. Any object that is not part of a cluster is categorized as noise. This is in contrast to *K-Means*, which assigns every object to a cluster. The DBSCAN algorithm works as follows: initially, all objects in the data set are assumed to be unassigned; DBSCAN then chooses an arbitrary unassigned object p from the data set; if DBSCAN finds p is a core object, it finds all the density-connected objects based on the input parameters and it assigns all these objects to a new cluster; if DBSCAN finds p is not a core object, then p is considered to be noise and DBSCAN moves onto the next unassigned object. Once every object is assigned, the algorithm stops.

The *K-Means* algorithm will be used in our classification methodology, mainly due to its simplicity and to the fact that it allows users to choose the number of clusters, allowing us to choose the number of studied applications.

4.3 Classification Methodology

The classification methodology uses two types of *data-streams*, defined in section 3.3.4, which are sampled in order to extract the following metrics:

- number of bytes and packets per sampling interval in the upload and download directions per sampling interval;
- number of bytes and packets per sampling interval in the download direction per sampling interval;
- number of bytes and packets per sampling interval in the upload direction per sampling interval.

Then, the multi-scaling analysis, presented in section 3.3.2, is performed to the obtained metrics using the tool available at [Dar08], as explained in section 3.3.3. In order to minimize the effects of the different values of available bandwidth, the q^{th} order spectrum estimators obtained from this analysis were normalized to zero mean. So, the normalized estimators $\hat{y}_{q,j}$ for the q-th order moment are computed as follows:

$$\hat{y}_{q,j} = y_{q,j} - \sum_{i=1}^{J} \frac{y_{q,j}}{J} \tag{4.2}$$

where J represents the maximum time scale considered for the q^{th} order spectrum. This allows us to observe the variation of the behavior patterns over the time scales, for the different traces, independently of the absolute values of the original data. We made this normalization since we want to differentiate applications based on the variations of their multi-scaling behavioral patterns and not based on their absolute values. Since both types of estimators, the ones obtained from the multi-scale decomposition of known traffic and of unknown traffic were normalized to zero mean, we use the notation introduced in (3.12) since it can extended to the different types of *data-streams*.

The normalized obtained decomposition estimators $\hat{y}_{q,j}$ were mixed and processed together in order to create clusters which will be labeled based on the assignment of the estimators obtained from the multi-scale decomposition of the *known data-streams*. We use an unsupervised clustering that provides a good and accurate cluster arrangement and where the number of clusters is pre-defined and equal to the different types of applications of the known flows. The clustering will group in the same cluster the spectrum estimators with similar behavior over the range of spectrum orders and considered decomposition scales. In this step, we used the *K-Means* algorithm, since it is one of the simplest and most efficient clustering techniques, besides allowing the choice of the number of clusters and always converging to a local optimum. The classification process based on the clustering algorithm is depicted in figure 4.1. Each normalized estimator is mapped into a J-dimensional data point and the clustering starts by randomly choosing k-points as the k-centroids of the initial clusters. The remaining data points, corresponding to the remaining estimators, are then assigned to the closest cluster. Subsequently, a new centroid is computed according to the performed assignments and if, after some iterations a converge criterion, which in our work included the stability of the centroids of each cluster as well the MSE, is met the algorithm stops. Otherwise, the data points are again assigned to the clusters according to the new centroids, each centroid is then recomputed and the convergence criterion is checked. When the algorithm stops, each cluster is assigned to the Internet application which has more estimators from *known data-streams* assigned to that cluster. All J-dimensional data-points, corresponding to the normalized estimators of the analyzed decomposition scales, are then classified. Moreover, it is an unsupervised technique, which is more appropriate for traffic classification since it does not rely on pre-defined classes. This is an important issue, since the profile of the training samples can be very different from the ones of the test samples, for the same group. At the end of this process, the created clusters obtained contain the known traffic streams, together with the unknown ones, allowing us to classify all traffic *data-streams* that can be further inputted to the classification tool. At the end, whenever possible and optionally, a validation

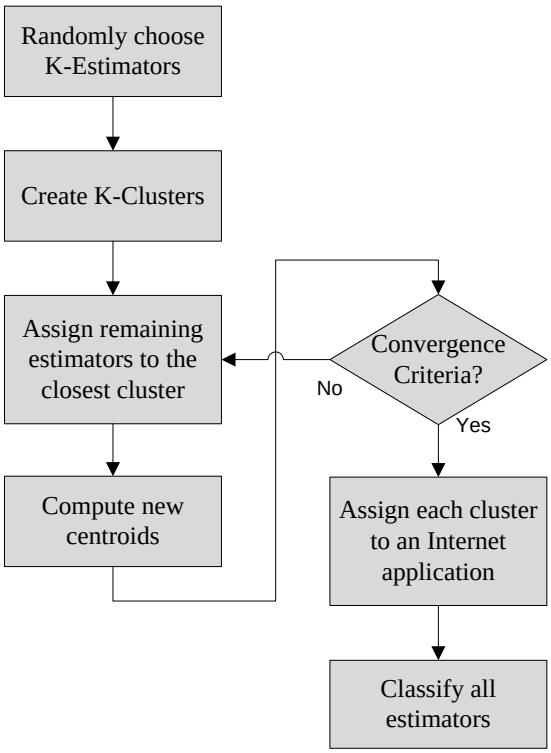

Figure 4.1: Flow diagram of clustering based classification methodology.

process (based on traditional identification techniques) can be performed in order to validate the classification and, if necessary, reclassify all traffic based on newly added known flows.

The proposed classification methodology can be adapted to a on-line procedure that relies on data obtained from an off-line procedure that is periodically executed. This is illustrated in figure 4.2 which presents a flow diagram that illustrates this off-line/on-line classification methodology. Such off-line procedure comprises the multi-scale decomposition of the sampled traffic metrics, the clustering of the the decomposition estimators and the association of each data-stream with an Internet application.

4.4 Classification Results

This section presents the results that were achieved and evaluates the performance and accuracy of the proposed classification methodology. Three distinct traffic statistics were considered for analysis in order to enable a comparative study on the differentiating capabilities of the different traffic data:

- number of transmitted bytes per sampling interval (independently of the traffic direction);

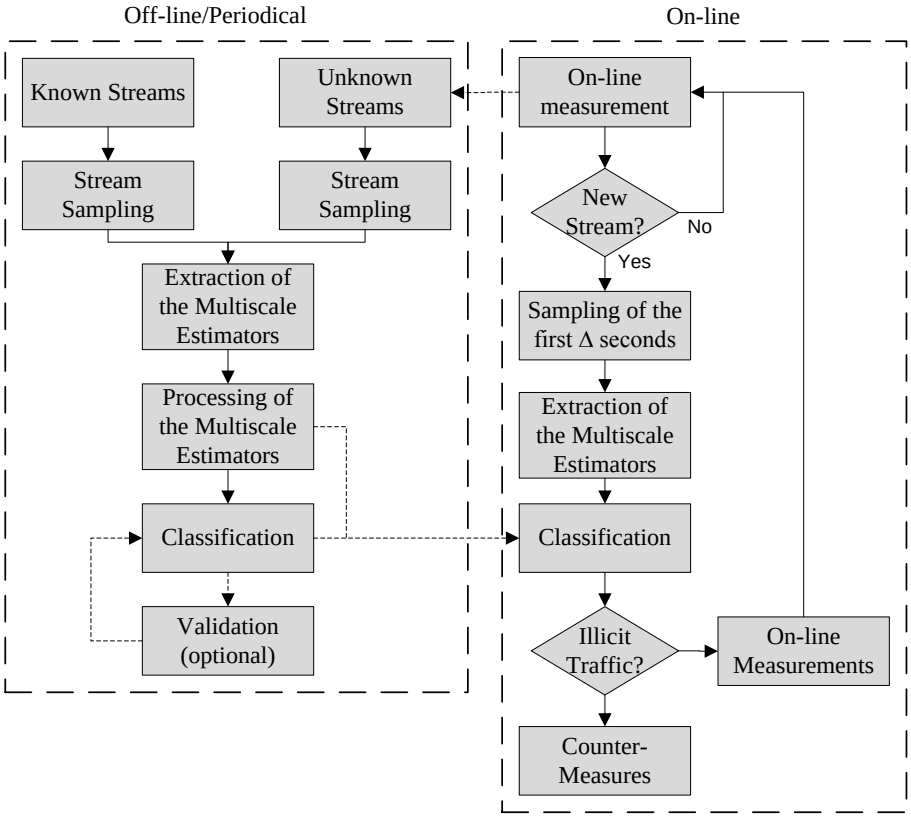

Figure 4.2: Flow diagram of the off-line and on-line classification methodology.

- number of downloaded bytes per sampling interval;
- number of uploaded bytes per sampling interval.

The sampling intervals used in this work were 100 *ms* and all flows were truncated to 30-minutes. Only the first two decomposition moments were considered for analysis ($Q = 2$) with the first decomposition scales ($J = 5$). Figure 4.3 shows the log-scale diagrams of the normalized first and second order wavelet spectrum decomposition estimators for the bidirectional (upload+download) traffic *streams* in the considered decomposition scales. As previously mentioned, the estimators in the log-scale diagram quantify the traffic scaling properties over the different temporal scales. Higher time scales are physically related to long-term actions, mainly at the user-level, such as user clicks over web page links or file download requests. Lower time scales are related to short-term interactions, such as packet/data generation and queuing or transmission control session dynamics. We have only used the first five decomposition scales for analysis, since at the higher scales the estimators tend to mix themselves. This is due to the fact that the long-term actions associated to these scales become similar for all applications. Several independent iterations (100) were performed in order to minimize

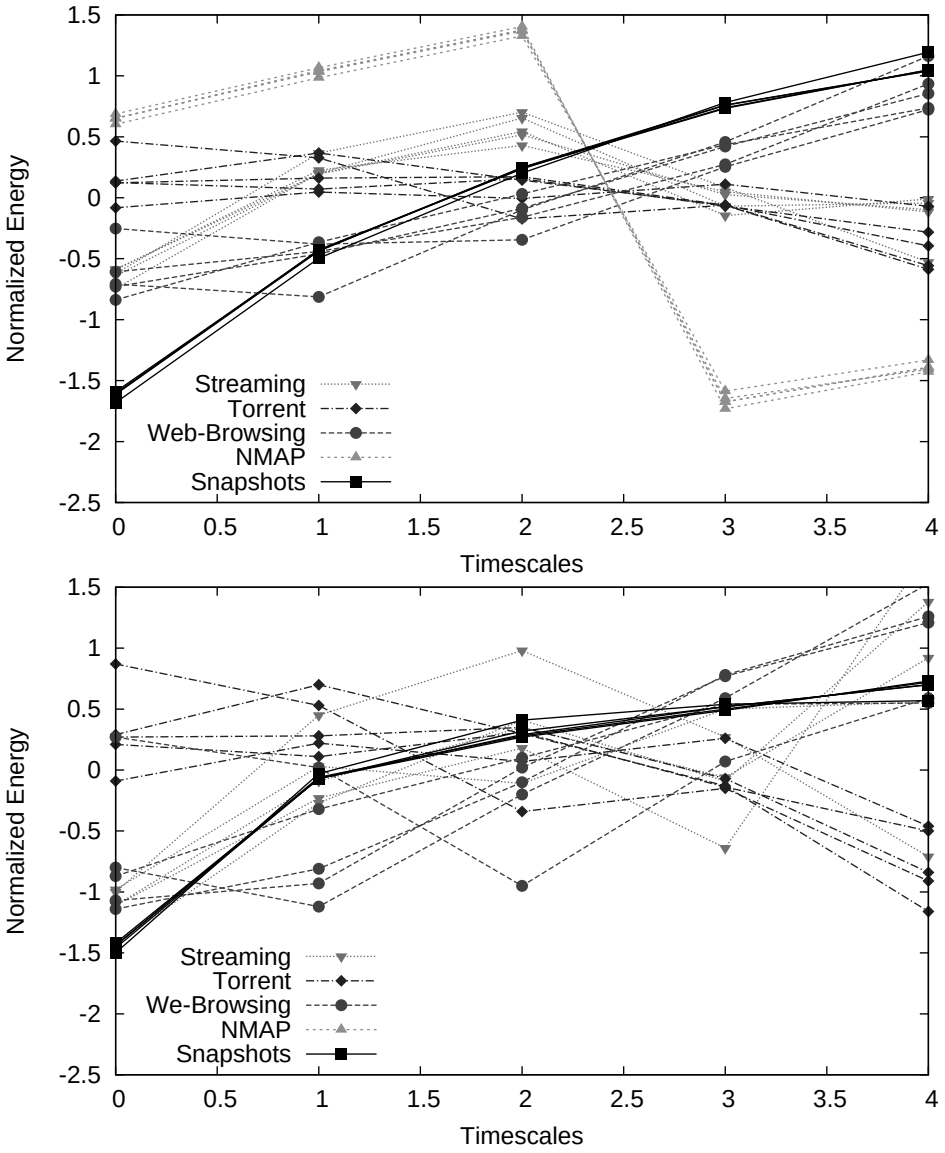

Figure 4.3: Normalized multi-scale estimators for the different upload+download traffic flows, (left) first order, (right) second order.

the effect of the initial choice of the random centroids.

As can be seen, for the first moment of the bidirectional profile there is a clear separation between all flows of the different Internet applications, which suggests that applying clustering analysis will lead to a very accurate separation of the analyzed *data-streams* on different clusters. In order to evaluate the accuracy of the proposed methodology, we used the classification of the *known data-streams* and verified if the different estimators obtained from the analysis of the *unknown data-streams* were correctly assigned to the cluster that was suggested/defined for each one of the selected protocols. Table 4.1 presents the classifi-

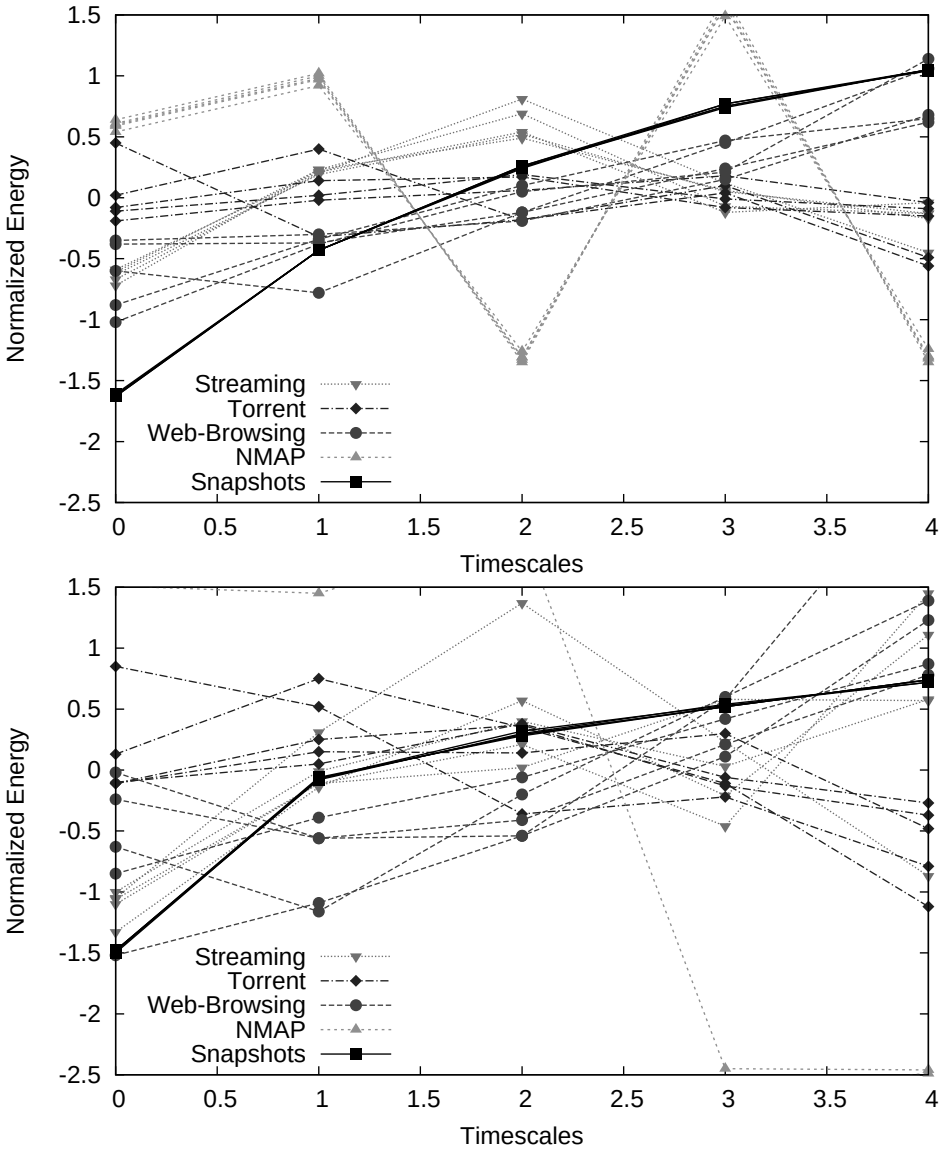

Figure 4.4: Normalized multi-scale estimators for the different download traffic flows, (left) first order, (right) second order.

cation results, using only the first order moment estimators, for the bidirectional profile. The classification accuracy is very high since all illicit traffic flows were correctly identified, as well as all WB data-streams. Some Torrent data-streams were wrongly assigned to the other applications, which can be explained by the nature of the P2P protocol that, when the client is connected to only one peer, reduces the overall communication to a client-server paradigm. Such paradigm can become similar to any other application whose profile consists in a simple client-server interaction. Some of the Video-Streaming streams were also classified as WB traffic, which can be explained by the fact that we have stream-alike transfers (for example,

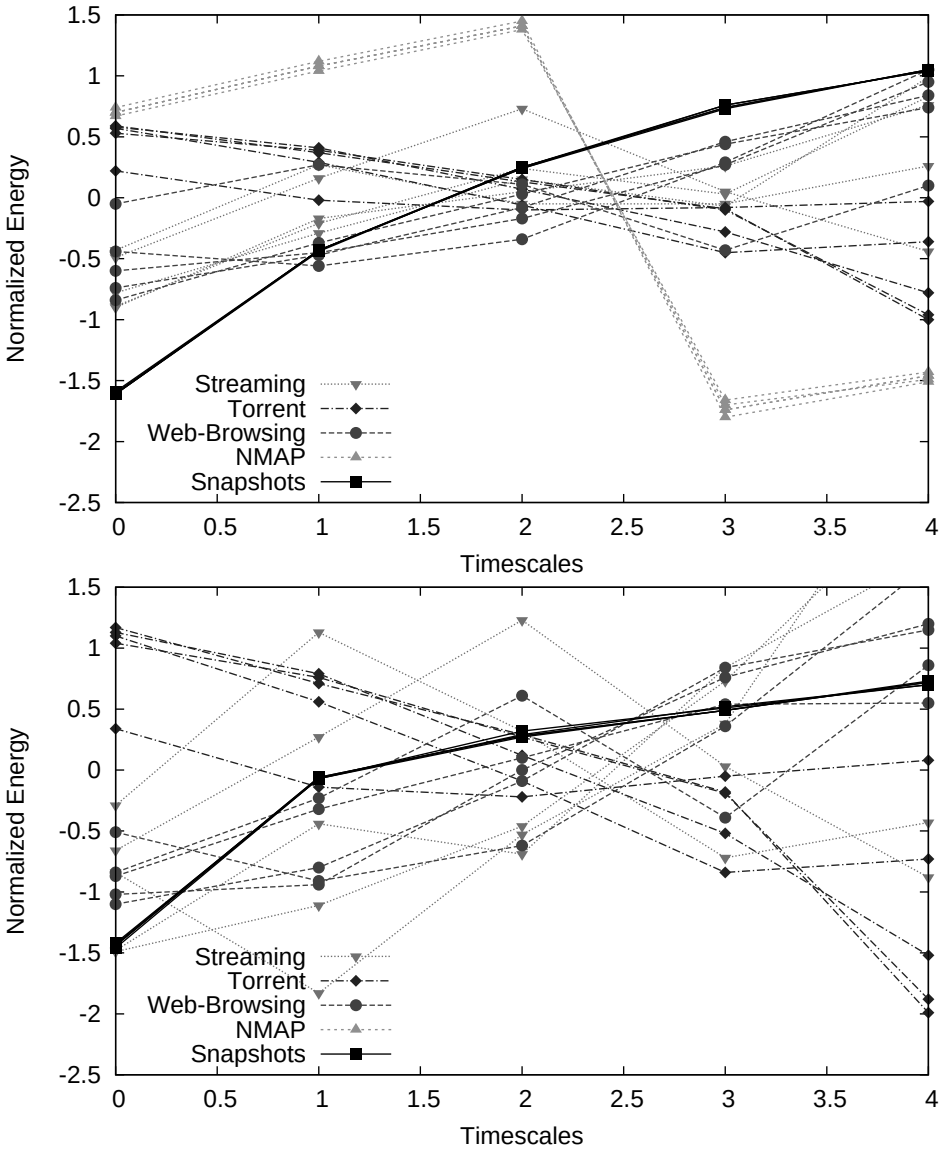

Figure 4.5: Normalized multi-scale estimators for the different upload traffic flows, (left) first order, (right) second order.

YouTube video downloads) embedded in HTTP communications. Therefore, it is possible to conclude that the proposed identification approach is able to accurately identify the three legitimate analyzed applications (Video-Streaming, BitTorrent and Web-Browsing) and was also able to identify 100% of the illicit traffic streams (port scan and snapshot).

When analyzing only the unidirectional profiles, whose log-scale diagrams are shown in figures 4.4 and 4.5, the accuracy decreases, as shown in tables 4.3 to 4.6. Indeed, for the first order normalized multi-scale estimators of the download case, the number of misclassified legitimate traffic streams increases. Indeed, the number of Web-Browsing *streams* assigned to

Table 4.1: Percentage of correctly classified *data-streams* for the upload+download traffic statistics.

	Using the first order moment				
	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**100%**	0%	0%	0%	0%
CI	100%-100%	0%-0%	0%-0%	0%-0%	0%-0%
Snapshots	0%	**100%**	0%	0%	0%
CI	0%-0%	100%-100%	0%-0%	0%-0%	0%-0%
Web-Browsing	0%	0.20%	**99.60%**	0%	0.20%
CI	0%-0%	0.07%-0.33%	99.20%-100%	0%-0%	0%-0.53%
Streaming	1.50%	1.80%	1.20%	**93.50%**	2.00%
CI	0.90%-2.10%	1.16%-2.44%	0.66%-1.74%	92.70%-94.30%	1.33%-2.67%
Torrent	1.90%	1.50%	2.70%	1.50%	**92.40%**
CI	1.25%-2.55%	0.90%-2.10%	1.3%-4.01%	1.00%-2.00%	90.10%-94.70%

other applications increases which can be explained by the fact that such *data-streams* may be originated by the visualization of an YouTube video, thus creating a download profile similar to the one corresponding to Streaming flows, or by the download of a large file available on a web-site, thus creating a profile similar to BitTorrent download traffic. On the other hand, the flows classified as Snapshot can be originated by the transfer of a file to an FTP server, which makes the profile of Web-browsing similar to the one corresponding to Snapshots. Most of the illicit flows were correctly identified, which confirms that traffic download statistics can be efficiently used for the identification of illicit traffic or of traffic presenting suspicious profiles.

Table 4.2: Percentage of correctly classified *data-streams* for the upload+download traffic statistics.

	Using the second order moment				
	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**100%**	0%	0%	0%	0%
CI	100%-100%	0%-0%	0%-0%	0%-0%	0%-0%
Snapshots	0%	**100%**	0%	0%	0%
CI	0%-0%	100%-100%	0%-0%	0%-0%	0%-0%
Web-Browsing	10.50%	7.50%	**62.60%**	12.00%	7.40%
CI	8.49%-12.51%	5.64%-9.36%	60.65%-64.55%	9.69%-14.31%	5.67%-9.13%
Streaming	20.40%	17.80%	18.30%	**21.40%**	22.10%
CI	17.39%-23.41%	14.72%-20.88%	15.55%-21.05%	20.65%-22.15%	18.84%-25.36%
Torrent	6.90%	6.10%	5.80%	5.80%	**75.40**
CI	4.87%-8.93%	4.17%-8.03%	3.91%-7.69%	3.87%-7.73%	73.74-77.06

Table 4.3: Percentage of correctly classified *data-streams* for the download traffic statistics.

	Using the first order moment				
	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**99.90%**	0%	0.10%	0%	0%
CI	99.73%-100.00%	0%-0%	0.03%-0.17%	0%-0%	0%-0%
Snapshots	3.00%	**88.00%**	5.00%	2.00%	2.00%
CI	0.15%-5.85%	82.39%-93.61%	1.36%-8.64%	0.66%-3.34%	0.87%-3.13%
Web-Browsing	3.50%	3.50%	**86.60%**	3.00%	3.40%
CI	2.38%-4.62%	2.43%-4.57%	85.32%-87.8%8	1.96%-4.04%	2.21%-4.59%
Streaming	3.40%	3.70%	3.10%	**86.30%**	3.50%
CI	2.61%-4.19%	2.89%-4.51%	2.33%-3.87%	84.95%-87.65%	2.57%-4.43%
Torrent	1.60%	3.80%	1.70%	3.50 %	**89.40%**
CI	0.99%-2.21%	1.84%-5.76%	0.70%-2.70%	2.06%-4.94%	87.07%-91.73%

Table 4.4: Percentage of correctly classified *data-streams* for the download traffic statistics.

	Using the second order moment				
	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**100%**	0%	0%	0%	0%
CI	100%-100%	0%-0%	0%-0%	0%-0%	0%-0%
Snapshots	0%	**100%**	0%	0%	0%
CI	0%-0%	100%-100%	0%-0%	0%-0%	0%-0%
Web-Browsing	4.90%	6.20%	**79.00%**	4.80%	5.10%
CI	3.71%-6.09%	4.98%-7.42%	77.03%-80.97%	3.73%-5.87%	4.09%-6.11%
Streaming	17.00%	16.70%	16.10%	**33.80%**	16.40%
CI	13.86%-20.14%	13.83%-19.5%7	12.93%-19.27%	29.80%-38.32%	13.19%-19.61%
Torrent	17.10%	19.00%	12.30%	17.00%	**34.60%**
CI	13.24%-20.96%	14.58%-23.42%	8.62%-15.98%	12.96%-21.04%	30.04%-39.10%

For the first order moments of the upload profile, the number of misclassified Web-Browsing and Video-Streaming streams increases, when compared to the previous cases. In fact, for Web-Browsing traffic,some streams were classified as Torrent, which can be caused by the transfer of a big file to an HTTP server that makes the upload profile of these flows to become similar to the Torrent profile. In addition, some streams were also assigned to Video-Streaming application which can due to some traffic generated when watching an embedded video on a web-page. The generated upload profile becomes similar to the one of Streaming applications since the client has to synchronize the video with the remote server. For the Video-Streaming application, the number of streams that were misclassified also increased, which is due to the similarity that exists between the Web-Browsing and Streaming upload

Table 4.5: Percentage of correctly classified *data-streams* for the upload traffic statistics.

	Using the first order moment				
	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**100%**	0%	0%	0%	0%
CI	100%-100%	0%-0%	0%-0%	0%-0%	0%-0%
Snapshots	10.00%	**68%**	4.00%	12.00%	6.00%
CI	4.99%-15.01%	63.04%-72.96%	0.73%-7.27%	6.58%-17.42%	2.04%-9.96%
Web-Browsing	9.70%	11.40%	**53.90%**	13.20%	11.80%
CI	7.42%-11.98%	8.86%-13.94%	51.24%-56.56%	10.38%-16.02%	9.07%-14.53%
Streaming	19.30%	21.30%	16.00%	**28%**	15.40%
CI	16.18%-22.42%	18.03%-24.57%	13.35%-18.65%	23.99%-32.01%	12.21%-18.59%
Torrent	0%	0.30%	0%	0%	**99.70**
CI	0%-0%	0.00%-0.60%	0%-0%	0%-0%	88.80%-93.40%

Table 4.6: Percentage of correctly classified *data-streams* for the upload traffic statistics.

	Using the second order spectrum				
	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**99.50%**	0.50%	0%	0%	0%
CI	99.00%-100.00%	0%-1.33%	0%-0%	0%-0%	0%-0%
Snapshots	1.00%	**98.00%**	0%	0%	1.00%
CI	0%-2.66%	96.00%-100.00%	0%-0%	0%-0%	0.33%-1.66%
Web-Browsing	6.70%	5.10%	**76.70%**	6.70%	4.80%
CI	5.38%-8.02%	3.84%-6.36%	75.38-78.02%	5.36%-8.04%	3.50%-6.10%
Streaming	2.20%	2.50%	4.30%	**86.4%**	4.60%
CI	1.30%-3.10%	1.55%-3.45%	3.02%-5.58%	81.97-90.83%	3.34%-5.86%
Torrent	2.20%	2.50%	4.30%	4.60%	**86.40%**
CI	1.30%-3.10%	1.55%-3.45%	3.02%-5.58%	3.34%-5.86%	85.21%-87.59%

profiles since both applications, on the client side, mostly send acknowledgments to signalize the reception of packets (Web-Browsing) or for synchronization purposes (video streaming). The classification accuracies for the BitTorrent and NMap applications were very high due to the unique profile of both applications. For BitTorrent, the fact the *peer* also uploads its files leads to the generation of significant traffic to the other connected *peers*. This creates a very distinct profile that our approach can accurately identify. For the NMap, the upload traffic consists of the SYN/ACK and SYN/RST packets sent to signalize that a requested port on the local host is either open or closed, respectively. This also creates an upload profile which is characteristic of the NMap application which is also accurately identified.

When analyzing the second order normalized decomposition estimators in all directions, the accuracy of our methodology decreases. These estimators account for the variance of the analyzed data, which can be seen as a measure of how spread out a distribution is. As can be seen from the analysis of tables 4.4 and 4.6, the accuracy of the methodology decreases for the licit traffic flows. This is due to the fact that the shape of the distributions of the traffic statistics become similar and, consequently, the distinction between the flows of the different protocols becomes less noticeable, leading to worse classification results.

The classification results were also computed for 5 and 15 minutes long traces. However, the classification accuracy when using such traces was considerably lower (between 30%-60% for all applications). Therefore, only the classification for the 30 minutes traces were presented.

4.5 Conclusions

In this chapter, we presented a first study that demonstrates that multi-scale traffic analysis is an accurate means to differentiate legitimate Internet applications and identify Internet attacks. The proposed methodology is based on clustering the multi-scale estimators inferred from different traffic profiles. By applying this framework to three licit applications (Web-Browsing, Streaming and BitTorrent) and two illicit applications (port scan and snapshot) that are very common on *botnets*, very accurate classification results were obtained using traffic statistics such as the number of bytes per sampling interval, independently of the traffic direction, and using only the first and second order multi-scale spectrum. Moreover, the identification was only based on the analysis of the first five time scales of the traffic for modeling the high-frequency components of the analyzed traffic.

A main drawback of this methodology is the fact that requires long traffic traces in order to obtain acceptable classification results, which compromises its suitability for pseudo real-time classification systems.

Chapter 5

Traffic Classification based on Probabilistic Modeling of the Traffic Multi-Scale Frequency Components

5.1 Introduction

This chapter presents several traffic classification approaches based on probabilistic models for associating unknown traffic to the corresponding Internet application. Our approaches can be divided in two main categories. The first comprises unidimensional probabilistic models in which each decomposition scale is analyzed separately. Classification is then performed by inferring an overall value that is obtained by considering the values at the several scales. The second comprises multidimensional probabilistic models in each the decomposition are jointly analyzed. Such distributions are inferred by mapping each decomposition scale to a dimension to generate a multidimensional space in which the probability distributions are generated. These distributions allow the analysis of the correlations between the values of the several decomposition scales in order to infer more accurate models.

These probabilistic methodologies enable an accurate traffic classification with short *data-streams* making them more suitable to be used in pseudo real-time classification systems, such as the architecture presented in section 1.3.

We start by presenting the several unidimensional approaches and subsequently the multidimensional ones. A section presenting, comparing and discussing the classification results is then provided followed by a discussion of the advantages and drawbacks associated to the several classification methodologies.

5.2 Unidimensional Probabilistic Modeling of the Decomposition Estimators

In this section we present our proposed classification methodologies based on the use of unidimensional probabilistic approaches for modeling the behavior of the multi-scale decomposition estimators. These methodologies constitute an enhancement to the approach presented in chapter 4, since the use of probabilistic classification methodologies enables a more accurate modeling of the different frequency components present in the analyzed Internet traffic. The different decomposition scales are evaluated separately and traffic is assigned to the Internet application whose distribution best fits its several frequency components.

This section starts by presenting some important background concepts and, then, a classification approach that models the first decomposition scales of the first order moment of analysis is presented. Then, an algorithm for the selection of the decomposition scales is presented, in order to enable more accurate classification results. Subsequently, two additional probabilistic modeling approaches are discussed: the first uses unidimensional Gaussian approaches to model the distributions generated by the estimators of the known data-streams of each Internet application, in a separate way for each decomposition scale; the second uses unidimensional generic probabilistic approaches. A discussion and comparison of the obtained results is then provided at the end of the chapter.

5.2.1 Background

The main concept of the approaches presented in this chapter, illustrated in figure 5.1, is the use of unidimensional probabilistic approaches for modeling the multi-scale decomposition estimators obtained from the analysis of the traffic of the different Internet applications. Such probabilistic models describe the behavior and the frequency components present in the analyzed traffic in the several decomposition scales.

The estimators obtained from the multi-scale decomposition of the known data-streams were used to validate the assumption that, for each application and scale, they can be modeled using Gaussian distributions. The Lilliefors goodness-of-fit test was used, allowing us to verify the null hypothesis that a sample in a vector comes from a distribution belonging to the Gaussian family, against the alternative that it does not [Lil67]. Assuming a random sample $X_1, X_2, \ldots, X_n$ of size n, the test proceeds as follows:

1. Compute the sample mean $\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$ for use as an estimate of μ;

2. Compute the sample variance $s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i - \overline{X})^2}$ as an estimate of σ;

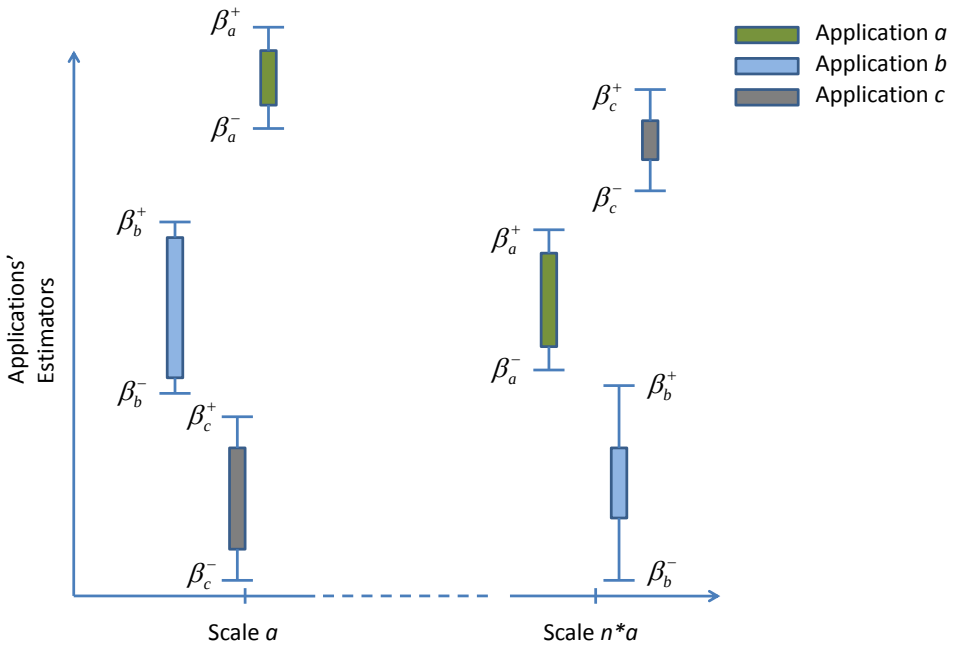

Figure 5.1: Unidimensional Probabilistic Modeling of the Multi-Scale Decomposition Estimators.

3. Compute the "normalized" sample values Z_i defined as $Z_i = \frac{X_i - \overline{X}}{s}, i = 1, \ldots, n$ from which the test will be computed;

4. Let $S(x)$ be the distribution function whose parameters are inferred from the samples Z_is. The Lilliefors test statistic T_1 is defined as: $T_1 = \sup_x |F^*(x) - S(x)|$ where $F^*(x)$ is the standard normal distribution function.

Two hypothesis result from the test:

- H_0 - this is the null hypothesis which determines that the samples come from a normal distribution;
- H_1 - the distribution function of the samples X_i is non-normal.

All the conducted tests did not reject the null hypothesis, that is, all the estimators can be approximated by a Gaussian distribution.

5.2.2 Choosing the decomposition scales

In this section, we present an algorithm that enhances our classification scheme. This algorithm (illustrated in figure 5.2) evaluates all the available decomposition scales contained in the set D, defined in (3.14), and selects the ones that can provide the best differentiation

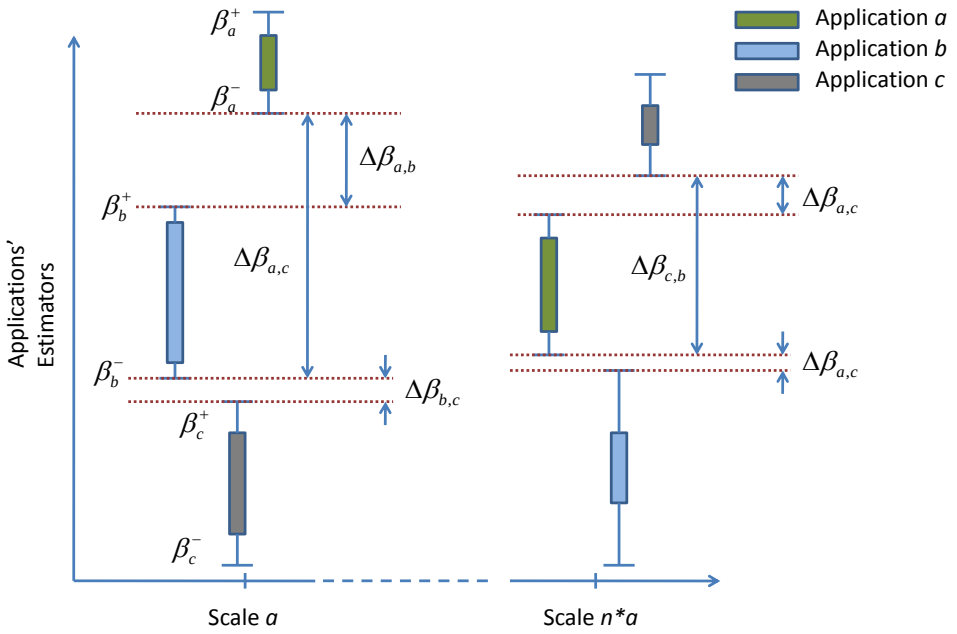

Figure 5.2: Algorithm for determining the best decomposition scales.

between applications. The main idea behind this approach is to determine the minimum distances between the distributions of the estimators of the different Internet applications in each scale of analysis, and use the ones where such distances are maximum. In this way, we are able to assure that the different distributions are well separated, which is essential in order to assure an accurate traffic discrimination.

In the following lines, the total number of decomposition scales, per Internet application, that will be used for analyzing all distributions is denoted by $\mathfrak{N}$. For all the available decomposition scales of all stochastic processes of all applications, we compute $\overline{e}_{a,\Gamma_z}, \beta^-_{a,\Gamma_z}$ and β^+_{a,Γ_z}, which represents the mean, the inferior and the superior quantiles, respectively, of the distribution inferred from the *known estimators* of an application a at time-scale j, moment q and stochastic process s, identified by Γ_z as defined in (3.13). These quantiles allow us to characterize the distributions inferred from the multi-scale estimators and determine their relative positions, for a specific decomposition scale, and can be defined as:

$$\begin{aligned} P[e^i_{a,\Gamma_z} < \beta^-_{a,\Gamma_z}] = \rho^-, \forall a \\ P[e^i_{a,\Gamma_z} < \beta^+_{a,\Gamma_z}] = \rho^+, \forall a \end{aligned} \tag{5.1}$$

where ρ^- and ρ^+ represent a inferior and a superior distributions thresholds defined *a priori*.

In order to quantify the (dis)similarity between the distributions inferred from the estimators of the different applications, we define, for all elements Γ_z of the set D, the following metric:

$$\Delta_{a,\alpha,\Gamma_z} = \begin{cases} \beta^-_{\alpha,\Gamma_z} - \beta^+_{a,\Gamma_z}, \bar{e}_{\alpha,\Gamma_z} > \bar{e}_{a,\Gamma_z} \\ \beta^-_{a,\Gamma_z} - \beta^+_{\alpha,\Gamma_z}, \bar{e}_{\alpha,\Gamma_z} \leq \bar{e}_{a,\Gamma_z} \end{cases} \tag{5.2}$$

with $a = 1, \ldots, A$ and $\alpha = 1, \ldots, A \wedge \alpha \neq a$, where it is assumed that the estimators of all elements Γ_z of the set D, defined in (3.14), of all N *known data-streams*, follow a generic probabilistic distribution with average

$$\bar{e}_{a,\Gamma_z} = \frac{1}{N} \sum_{i=1}^{N} e^i_{a,\Gamma_z} \tag{5.3}$$

We can then define as (dis)similarity metric for each application and for each process, moment and dimension defined by Γ_z, the minimum distance between the estimators distribution of one application $a, a = 1, \ldots, A$, to all the remaining ones. This can be formulated as follows:

$$d_{a,\Gamma_z} = \min_{\alpha}[\Delta_{a,\alpha,\Gamma_z}], \forall \alpha \neq a \tag{5.4}$$

The set $C, |C| = L$, which refers to the set of the chosen dimensions, for all applications can subsequently be defined. In addition, let

$$\zeta_l = \{\eta_l, \nu_l, \gamma_l\} \in \mathbb{N}^3 \tag{5.5}$$

represent the $l-th$ element, indexing a stochastic process η_a, moment ν_a and scale γ_a, of the set C such that, for all elements of the set D, the L-distances d_{a,Γ_z} are higher, *i.e*,

$$C = \{\zeta_l | d_{a,\zeta_l} = \max_{z}^{n}[d_{a,\Gamma_z}]\} \tag{5.6}$$

with $a = 1, \ldots, A$, $n = 1, \ldots, \mathfrak{N}$, $l = 1, \ldots, L$ where $L = \mathfrak{N}A(a-1) + n$ and $\max_{z}^{n}[d_{a,\Gamma_z}]$ represents the n-th maximum distances.

5.2.3 Classification using unidimensional Gaussian distributions

In this section, we present a classification methodology based on the assumption that the multi-scale estimators for a specific application process within the same scale (of the same moment) follow a Gaussian distribution with mean $\bar{e}_{a,s,q,j}$, inferred as defined in (5.3), and (not-null) variance $\sigma^2_{a,s,q,j}$ inferred as:

$$\sigma^2_{a,\Gamma_z} = \frac{1}{N-1} \sum_{i=1}^{N} (e^i_{a,\Gamma_z} - \bar{e}_{a,\Gamma_z})^2. \tag{5.7}$$

with $s = 1, \ldots, S$, $q = 1, \ldots, Q$ and $j = 1, \ldots, J$.

The analysis is performed over the decomposition scales that were selected using the

algorithm described in section 5.2.2. For each of the selected decomposition scales, the distribution generated by the estimators of the known data-streams of each application are then used to determine to which distribution describes best the frequency components present in the corresponding decomposition scale. Therefore, let the probability value P_{i,a,ζ_l} which is computed as:

$$P_{i,a,\zeta_l} = \frac{e^{\left(\frac{-(u^i_{\zeta_l} - \overline{e}_{a,\zeta_l})^2}{2\sigma^2_{a,\zeta_l}}\right)}}{\sqrt{2\pi\sigma^2_{a,\zeta_l}}} du \tag{5.8}$$

with $i = 1, \ldots, M$, $a = 1, \ldots, A$, represent the probability that a multi-scale quantifier of the stochastic process η_a of an *unknown data-stream* i, at scale γ_a from moment ν_a indexed by ζ_l, belongs to the Gaussian distribution of the estimators of application a in the same time-scale.

Let $P_{i,a}, i = 1, \ldots, M, a = 1, ..., A$ designate the probability that the *unknown data-stream* i is originated by application a. This probability will be estimated by averaging the partial probabilities, defined in (5.8), over the time scales of better differentiation defined by ($\{\zeta_l = (\eta_l, \nu_l, \gamma_l), l = 1, \ldots, L\}$):

$$P_{i,a} = \frac{1}{L} \sum_{l=1}^{L} P_{i,a,\zeta_l}. \tag{5.9}$$

Finally, an *unknown data-stream* i is associated with an application α whose distribution maximized the previously computed value, *i.e*

$$\exists \alpha, P_{i,\alpha} = \max_a [P_{i,a}] \tag{5.10}$$

for $i = 1, \ldots, M$, $a = 1, \ldots, A$ and $\alpha = 1, \ldots, A$.

5.2.4 Classification using unidimensional generic distributions

In this section, we present a classification methodology based on the assumption that the multi-scale estimators for a specific application process within the same scale (of the same moment) follow a generic distribution effectively characterized by an average value $\overline{e}_{a,s,q,j}$ and quantiles $\beta^-_{a,s,q,j}$ and $\beta^+_{a,s,q,j}$ at thresholds ρ^- and ρ^+, respectively. Therefore, we use the quantiles as defined in (5.1) and the average as defined in (5.3) to compute a metric (of distance) $\Gamma_{i,a,\eta_a,\nu_a,\gamma_a}$ between a multi-scale quantifier of the *unknown data-stream* i and the general distribution of the multi-scale estimators of stochastic process s of application a in scale j of moment q. This distance metric is calculated over the set of selected time scales previously identified, and defined by $\{\zeta_l = (\eta_l, \nu_l, \gamma_l), l = 1, \ldots, L\}$:

$$\Gamma_{i,a} = \frac{1}{L}\sum_{l=1}^{L}\left|\beta^{+}_{a,\zeta_l} + \beta^{-}_{a,\zeta_l} - 2u^{i}_{\zeta_l}\right| \tag{5.11}$$

with $i = 1, \ldots, M$ and $a = 1, \ldots, A$.

Finally, an *unknown data-stream* i is associated with an application α such that

$$\exists \alpha, \Gamma_{i,\alpha} = \min_{\alpha}[\Gamma_{i,a}]. \tag{5.12}$$

for $i = 1, \ldots, M$, $a = 1, \ldots, A$ and $\alpha = 1, \ldots, A$.

5.3 Multidimensional Probabilistic Modeling of the Decomposition Estimators

This section presents probabilistic approaches that use multivariate and multidimensional distributions to model the behaviors and frequency components present in each scale of analysis. The use of multidimensional approaches allows the analysis of the correlations between the estimators at the different scales of analysis, which enables the use of more accurate distributions. A more accurate identification and discrimination of the traffic generated by the different Internet applications, as well as the identification of illicit traffic can be achieved.

5.3.1 Background

The main concept behind the methodologies presented in this chapter is the use of multidimensional distributions. A multidimensional distribution is inferred for each studied Internet application, as illustrated in figure 5.3. Unknown traffic is then classified based on the distributions generated by the estimators obtained from the multi-scale decomposition.

A vector $x = [x_1, x_2, \ldots, x_n]^T$ is said to have a Gaussian distribution with mean $\mu \in \mathbb{R}^n$ and covariance matrix Σ if its probability density function is given by [Do08]

$$P_{(x;\mu,\sigma)} = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \tag{5.13}$$

The coefficient $\frac{1}{(2\pi)^{n/2}}$ is used as an energy preservation factor used to ensure that

$$\frac{1}{(2\pi)^{n/2}}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{+\infty}\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)dx_1dx_2\ldots dx_n = 1 \tag{5.14}$$

When working with multiple variables, the covariance matrix provides a succinct way to analyze the correlations between all pairs of variables.

There are several advantages associated to the usage of such approaches. For example, in cases where the analyzed data has a high dimensionality or complexity, multidimensional

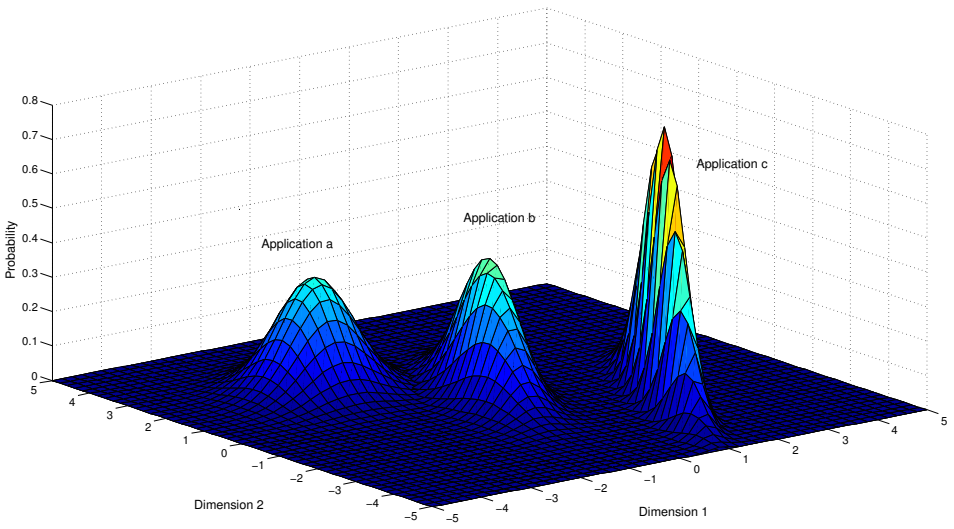

Figure 5.3: Multi-scale estimators for the different stochastic processes of sampled data-streams.

approaches are adequate for modeling and providing a single output value. Besides, the use of these approaches allows the study of the correlation between the values of the different variables, which leads to more accurate distributions.

5.3.2 Classification using Multidimensional Gaussian Approaches

This sub-section presents a classification methodology that is based on the assumption that the multi-scale estimators, for a specific application, follow a multivariate Gaussian distribution. Thus, let us define, for each application, a $L-$vector of mean values $\vec{e_a} = (\bar{e}_{a,\zeta_l}), \forall l = 1, \ldots, L$, and a $L-$vector of variances $\vec{\sigma_a^2} = (\sigma^2_{a,\zeta_l}), \forall l = 1, \ldots, L$, which can be inferred as:

$$\bar{e}_{a,\zeta_l} = \frac{1}{N} \sum_{i=1}^{N} e^i_{a,\zeta_l}, \forall l \tag{5.15}$$

$$\sigma^2_{a,\zeta_l} = \frac{1}{N-1} \sum_{i=1}^{N} (e^i_{a,\zeta_l} - \bar{e}_{a,\zeta_l})^2, \forall l \tag{5.16}$$

Therefore, let us define the probability that the multi-scale estimator u^i_C obtained from the multi-scale decomposition of a stochastic process of the $i-$th *unknown data-stream*, using the set of chosen dimensions C defined in (5.6), belongs to the Gaussian distribution whose parameters are inferred from the estimators obtained from the multi-scale decomposition of the *known data-streams* of an application a, using the same set of dimensions, as:

$$P_{i,a} = \frac{1}{(2\pi)^{L/2}|\Sigma_{a,C}|^{1/2}} e^{-\frac{1}{2}(u_C^i - \vec{e_a})^T \Sigma_{a,C}^{-1}(u_C^i - \vec{e_a})} \tag{5.17}$$

$\forall a = 1, \ldots, A, \forall i = 1, \ldots, M$, where $\vec{e}_a$ is a L-vector and $|\Sigma_{a,C}|$ is the determinant of $\Sigma_{a,C}$, which is the $L \times L$ covariance matrix in the set of dimensions indexed by C.

Finally, an *unknown data-stream* i is associated with an application α such that

$$\exists \alpha, P_{i,\alpha} = \max_a [P_{i,a}] \tag{5.18}$$

for $i = 1, \ldots, M$, $a = 1, \ldots, A$ and $\alpha = 1, \ldots, A$.

5.3.3 Classification using Multidimensional Generic Approaches

Let us now assume that the distributions inferred from the estimators of the *known data-streams*, of each studied application, follow a generic multi-dimensional distribution. Therefore, the estimator u_C^i obtained from the multi-scale decomposition of a stochastic process of the $i-$th *unknown data-stream*, using the set of chosen dimensions C defined in (5.6), will be classified according to the distance $D_{i,a}, a = 1, \ldots, A$ to the distribution corresponding to the Internet application a. The distance used in this work is the Mahalanobis distance since it considers the correlations between the different variables and can be defined as follows [Mah36]:

$$D_{i,a} = \sqrt{(u_C^i - \vec{e_a})^T \Sigma_{a,C}^{-1}(u_C^i - \vec{e_a})}, \forall a \tag{5.19}$$

where $\Sigma_{a,C}^{-1}$ is the covariance matrix in the set of dimensions indexed by C. The i-th unknown *data-stream* will then be associated to the distribution of the application α that minimizes the distance defined in (5.19), *i.e*:

$$\exists \alpha, D_{i,\alpha} = \min_a \left[D_{i,a}\right], a = 1, \ldots, A \tag{5.20}$$

5.4 Classification Results

This section presents the identification results obtained from applying the previously presented classification methodologies. First, we present the results obtained when classifying the *streams* using unidimensional probabilistic approaches, as described in section 5.2. Subsequently, the results obtained when using multidimensional probabilistic approaches, as described in section 5.2, are presented.

The identification methodologies were applied to traffic *data-streams* extracted from: (i) licit TCP and UDP traffic traces passively collected at the University of Aveiro network on September 15, 2008 and (ii) illicit traces that were experimentally generated in our laboratory

in order to simulate some of the most relevant *botnet* uses. The total number of considered applications was 5 ($A = 5$), 3 licit and 2 illicit. The licit applications *data-streams* that were extracted (and classified *a priori*) from the collected traffic belong to file-sharing (BitTorrent), Video Streaming and Web-Browsing. On the other hand, the illicit applications (NMap and Snapshots) were generated with the profile described in section 3.2.2. The complete dataset is composed by 50 *data-streams* of each application.

5.4.1 Unidimensional Probabilistic Approaches

Without selecting decomposition scales

In this section we present the classification results obtained using unidimensional Gaussian distributions only, as described in section 5.2.3, without using the selection scales algorithm. Therefore, we consider the first moment ($Q = 1$) and first five decomposition scales ($J = 5$). The analyzed metric is the number of captured bytes sent and received by a local host ($S = 1$).

The classification results were computed by comparing the classification achieved with the proposed methodology with the real applications of the *data-streams*, that are known *a priori*. We considered 5 and 15 minutes long *data-streams*. We only used the first 5 scales, since the estimators of all applications tend to converge at higher scales. Figures 5.4 and 5.5 show box plots with the 25%, 50%, 75% and 95% quantiles of the estimators of the first order moment of the normalized multi-scale decomposition estimators corresponding to 5 minutes and 15 minutes *data-streams*, respectively. From these plots, we can observe that the distributions of the estimators of the Web-Browsing and Snapshot *streams* almost overlap in all scales. This suggests that some Web-Browsing and Snapshot *streams* might be misclassified. However, for the 15 minutes *data-streams* (Figure 5.5) the Snapshot traffic estimators are now more concentrated around the mean, which suggests that the accuracy will be higher. For the remaining estimators' distributions, we can observe that, at least in one scale, they are very separated and therefore they should not be misclassified.

The numerical results obtained, for the 5 minutes traffic traces, are presented in Table 5.4.1. It is possible to observe that the results obtained are relatively accurate for all applications with a percentage of correctly identified *data-streams* between 72% and 100%. For Web-Browsing traffic, the correct classification percentage is lowest, as some of these *data-streams* were misclassified as Snapshot, which is in accordance with the previous analysis. This result can also be explained by the fact that the multi-scale estimators of the Web-Browsing *data-streams* have an higher variance, resulting from the various and heterogeneous user behaviors, making this distribution to partially overlap with the distribution of the snapshot estimators (which has a much lower variance) at all scales. Moreover, several protocols, such as file sharing and video streaming, run on top of Web-Browsing communications, which justifies the large variance that the estimators of these *streams* present and some of the classification mistakes that are obtained.

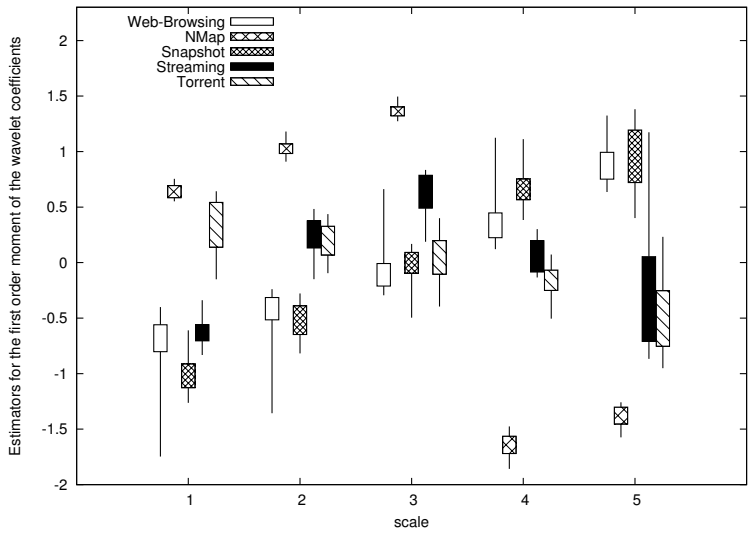

Figure 5.4: Distributions for first order decomposition estimators of the 5 minutes traces of the studied Internet applications and attacks.

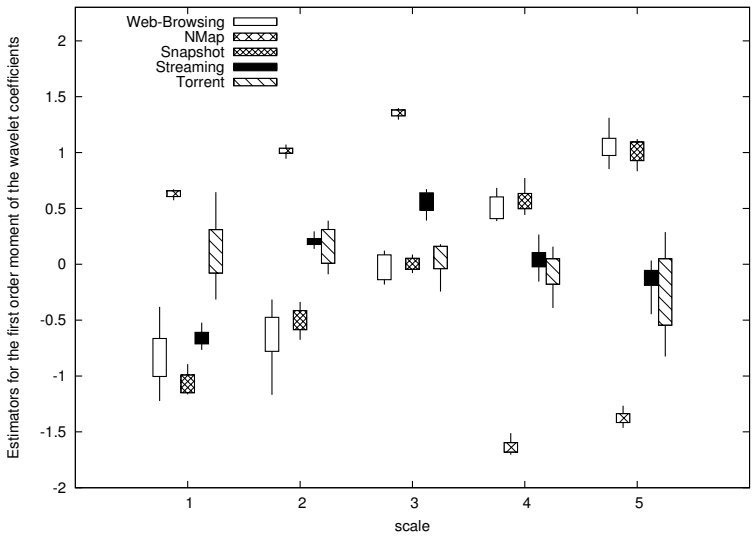

Figure 5.5: Distributions for first order decomposition estimators of the 15 minutes traces of the studied Internet applications and attacks.

The classification results for the 15 minutes *data-streams* are presented in Table 5.4.1, where we can observe that the accuracy of the results for all applications is higher. This can be explained by the fact that since traces are longer, they contain more information and more differentiating characteristics of the different applications. This obviously allows a more accurate decomposition of each data-stream and, therefore, a better analysis of the several frequency components, leading to better classification results.

Table 5.1: Percentage of correctly classified *data-streams* for the first order moment using 5 minutes traces.

	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**100%**	0%	0%	0%	0%
CI	100%-100%	0%-0%	0%-0%	0%-0%	0%-0%
Snapshots	0%	**74.01%**	23.89%	1.89%	0.21%
CI	0%-0%	69.92%-78.10%	19.61%-28.17%	1.47%-2.31%	0.08%-0.33%
Web-Browsing	0%	23.65%	**72.39%**	1.16%	2.80%
CI	0%-0	20.53%-26.78%	69.39%-75.39%	0.82%-1.50%	2.37%-3.23%
Streaming	0%	1.91%	5.54%	**92.52%**	0.03%
CI	0%-0%	1.42%-2.40%	4.90%-6.20%	91.85%-93.18%	0%-0.09%
Torrent	0%	0.41%	3.68%	0.07%	**95.84%**
CI	0%-0%	0.07%-0.74%	2.93%-4.44%	0.05%-0.09%	94.92%-96.76%

Table 5.2: Percentage of correctly classified *data-streams* for the first order moment using 15 minutes traces.

	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**100%**	0%	0%	0%	0%
CI	100%-100%	0%-0%	0%-0%	0%-0%	0%-0%
Snapshots	0%	**95.34%**	4.66%	0%	0%
CI	0%-0%	94.00%-96.67%	3.33%-6.00%	0%-0%	0%-0%
Web-Browsing	0%	15.48%	**74.52%**	0.22%	9.78%
CI	0%-0%	12.72%-18.23%	71.47%-77.58%	0%-0.48%	8.24%-11.32%
Streaming	0%	0%	0.09%	**98.48**	1.430%
CI	0%-0%	0%-0%	0%-0.25%	97.94%-99.01%	0.91%-1.95%
Torrent	0%	0%	1.49%	0%	**98.51%**
CI	0%-0%	0%-0%	0.59%-2.38%	0%-0%	97.62%-99.41%

Using selected decomposition scales

Let us now present the results obtained when using the algorithm for identifying the decomposition scales more suitable for an accurate traffic identification, presented in section 5.2.2. The considered stochastic processes were the byte counts per sampling interval (0.1 seconds) in the download ($s = 1$) and upload directions ($s = 2$) and the packet counts per sampling interval (0.1 seconds) in the download ($s = 3$) and upload directions ($s = 4$). These estimators were computed for the first eight time-scales ($J = 8$) of the first three order moments ($Q = 3$).

Figure 5.6 shows an example of some of the obtained decomposition estimators, as defined in (3.12), for all the available stochastic processes, moments and time-scales. We can observe that estimators for *data-streams* of the same application have a strong similarity, while *data-streams* of different applications exhibit relative dissimilarity. This suggests that an accurate

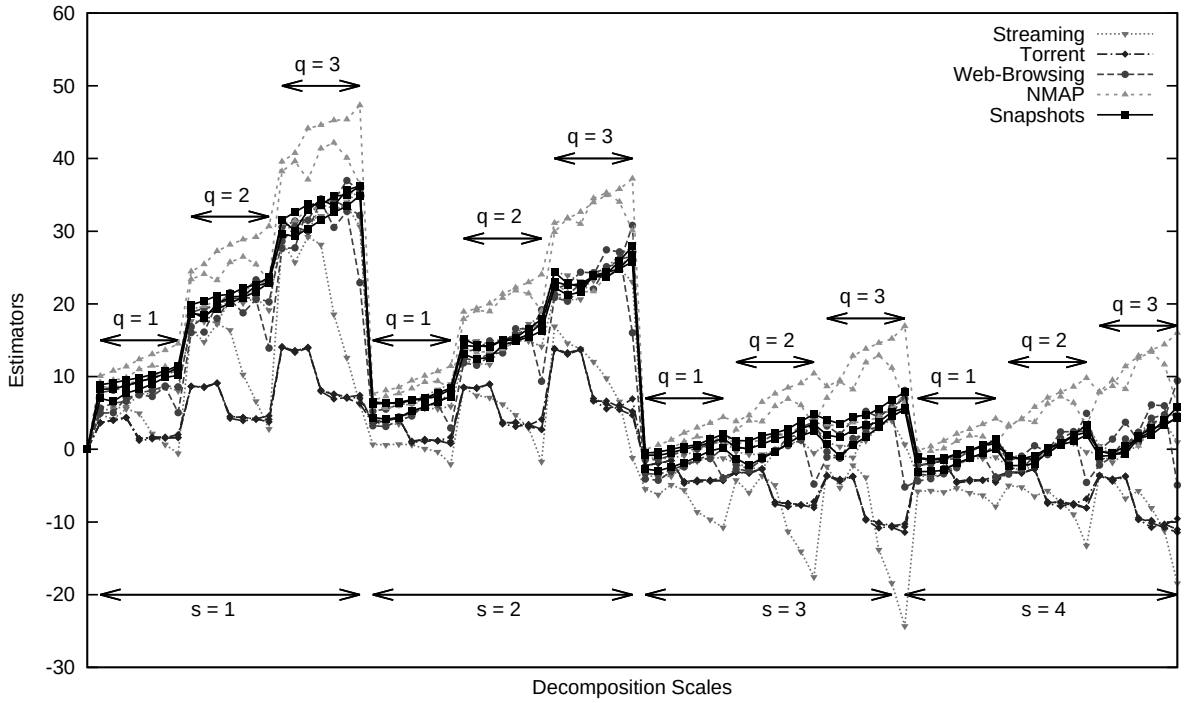

Figure 5.6: Multi-scale estimators for the different stochastic processes of sampled data-streams.

Table 5.3: Percentage of correctly classified *data-streams* using a unidimensional generic distribution

	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**100%**	0%	0%	0%	0%
CI	100%-100%	0%-0%	0%-0%	0%-0%	0%-0%
Snapshots	0%	**100%**	0%	0%	0%
CI	0%-0%	100%-100%	0%-0%	0%-0%	0%-0%
Web-Browsing	0%	0%	**94.13%**	5.87%	0%
CI	0%-0%	0%-0%	92.72%-95.55%	4.45%-7.28%	0%-0%
Streaming	0%	0%	10.40%	**55.47%**	34.13%
CI	0%-0%	0%-0%	8.99%-11.81%	52.92%-58.02%	31.53%-36.73%
Torrent	0%	1.46%	0%	0.07%	**98.47%**
CI	0%-0%	0.67%-2.26%	0%-0%	0%-0.02%	97.67%-99.26%

classification of the traffic *streams* is possible. Moreover, the flows originated by the illicit activities present very particular behaviors, caused by the nature of the traffic *data-streams*, that differs from the licit ones. This indicates that an accurate identification of such illegal activities will be achieved.

Table 5.4: Percentage of correctly classified *data-streams* using a unidimensional Gaussian distribution

	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**100%**	0%	0%	0%	0%
CI	100%-100%	0%-0%	0%-0%	0%-0%	0%-0%
Snapshots	0%	**91.53%**	1.47%	4.67%	2.33%
CI	0%-0%	89.50%-93.57%	0.86%-2.08%	2.99%-6.35%	1.35%-3.31%
Web-Browsing	4.40%	1.80%	**92.80%**	1.00%	0%
CI	3.47%-5.33%	0.78%-2.82%	91.38%-94.22%	0.48%-1.52%	0%-0%
Streaming	0%	1.20%	8.87%	**77.26%**	12.67%
CI	0%-0%	0.19%-2.21%	7.57%-10.16%	74.63%-79.88%	10.49%-14.84%
Torrent	0%	1.93%	0%	5.07%	**93.00%**
CI	0%-0%	0.83%-3.03%	0%-0%	3.78%-6.35%	91.88%-94.12%

Table 5.5: Percentage of correctly classified *data-streams* using unidimensional generic and Gaussian distributions.

	Generic distribution methodology Identified as		Gaussian distribution methodology Identified as	
	Licit traffic	**Illicit traffic**	**Licit traffic**	**Illicit traffic**
Licit traffic	**99.51%**	0.49%	**96.89%**	3.11%
CI	99.25%-99.77%	0.23%-0.75%	96.24%-97.54%	2.47%-3.75%
Illicit traffic	0%	**100%**	4.23%	**95.77%**
CI	0%-0%	100%-100%	3.22%-5.24%	94.75%-96.79%

Tables 5.4.1 to 5.4.1 show the identification results obtained after the completion of the 100 independent experiments, together with the corresponding 95% CI. The results include, for both methodologies, the identification performance when identifying (i) individual applications and (ii) groups of licit/illicit applications. This last set of results is much more relevant to a network traffic data analysis with the purpose of identifying the source(s) of illicit activities in a network.

We can see that the classification accuracy in both methodologies is very high for all studied applications, except for the Video-Streaming application. This result is explained by the fact that the traffic generated by the Video-Streaming application reveals a near-constant bandwidth utilization that includes some pseudo-periodic short duration bursts associated with pseudo-periodic periods of bandwidth starvation. These characteristics make the streaming profile similar, in some parts, to Web-Browsing (pseudo-periodic short bursts) and, in other parts, similar to BitTorrent (high bandwidth utilization, but with less variation), making the decision process complex and susceptible to errors. However, it is important to notice that both methodologies performed extremely well when identifying illicit traffic as a group. The methodology that assumes a generic distribution of the estimators had a perfect score of

classification. Nevertheless, the methodology where a generic distribution of the multi-scale estimators is assumed performed better when identifying and differentiating individual applications. The proposed identification methodologies also returned a very small percentage of false positives when identifying illicit traffic. These values make both methodologies very interesting tools for network security purposes, since the margin of error is acceptable in such complex, and many times, restricted environments.

5.4.2 Multidimensional Probabilistic Approaches

In this section, the results obtained with the multidimensional classification methodologies are presented and discussed. The classification methodologies were applied to the *data-streams* presented in section 3.2.2. The stochastic processes considered in this work are the bytes and packets counts in the upload and download directions, per sampling interval of 0.1 seconds ($S = 4$). The analysis was performed for the first three order moments ($Q = 3$), while the number of chosen dimensions, per application, was equal to 2 ($\mathfrak{N} = 2$) and the total number of used dimensions was 10 ($L = 10$).

Before inferring the multi-dimensional distributions, we had to verify if the estimators of the known *data-streams*, for each dimension, followed a Gaussian distribution or not. The Lilliefors goodness-of-fit test was used, which verifies the null hypothesis that a vector sample comes from a distribution of the Gaussian family, against the alternative that it does not [Lil67]. All the tests did not reject the null hypothesis, that is, all the estimators, for each dimension, can be approximated by a Gaussian distribution. Figures 5.4.2 and 5.7 show some of the estimators obtained from the multi-scale decomposition of the *known data-streams* of the studied protocols. For illustration purposes, we only show three dimensions and three applications per figure. We can see that the estimators obtained from *streams* of the same application are positioned in the same region and are separated from the ones of the remaining applications on the same multi-dimensional space. We can also assume that, by using more dimensions per application, these differentiations will increase. Thus, the distributions inferred from these estimators will be very differentiable, which results in a good discrimination between the traffic of the different applications. This suggests that the results will present a very high accuracy. However, we can see that some of the estimators of the Streaming traffic are somehow spread and some of them even approximate the estimators of the Web-Browsing protocol. This suggests that some of the Streaming traffic can be classified as Web-Browsing traffic.

In order to evaluate the accuracy of the proposed methodology and to assess the influence of the random choice of the known traffic *streams*, 100 independent iterations were run. In each iteration, 20 *streams* were randomly chosen as *known streams* ($N = 20$), while the remaining ones were used as *unknown streams* ($M = 30$) and classified in order to evaluate the efficiency of the proposed methodology. The threshold values ρ^- and ρ^+ were set to 5%

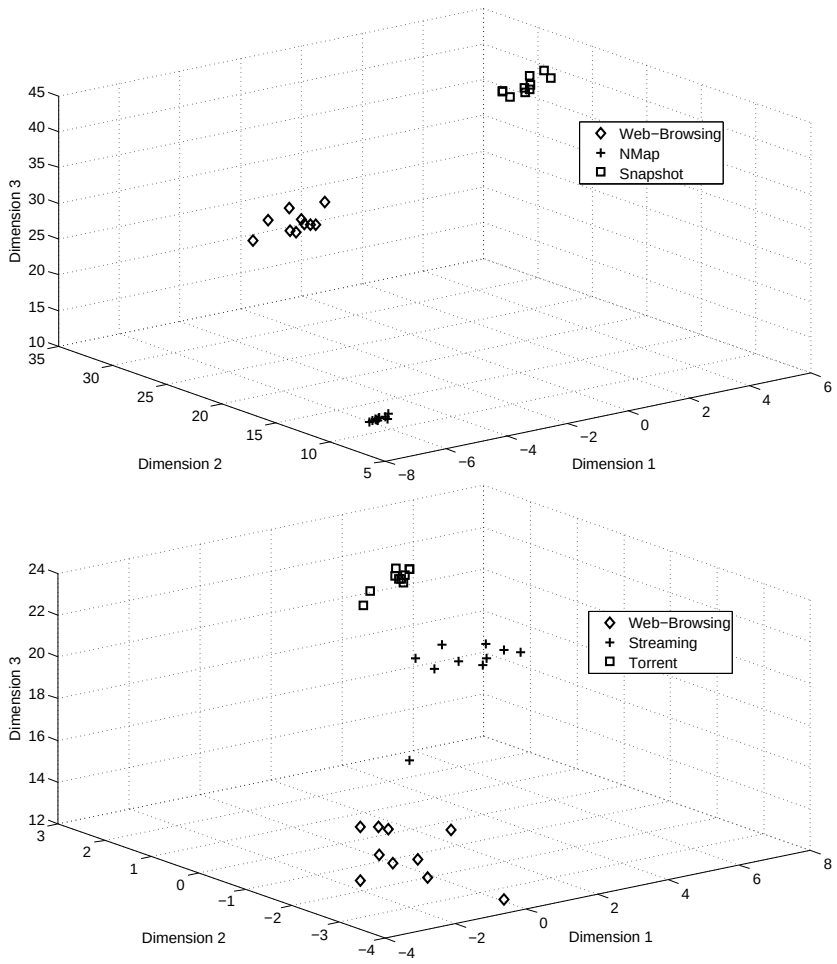

Figure 5.7: Sample Estimators extracted from sample Web-Browsing, NMap and Snapshot *streams* (top) and Sample Estimators extracted from sample Web-Browsing, Streaming and BitTorrent *streams* (bottom).

and 95%, respectively.

Table 5.6 shows the identification results obtained, together with the corresponding 95% CI, when using multidimensional Gaussian approaches. We can see that the percentage of correctly classified flows is always higher than 91%. For the illicit traffic, the classification rates are equal to 100% and 97.23%, for the NMap and Snapshot *streams* respectively. This shows that these attacks can be identified with very high accuracy using this methodology. For the Snapshot case, some of the *streams* were erroneously classified as Web-Browsing traffic, which can be explained by the fact that the Snapshot traffic was emulated through the upload of a desktop image every time the user performed a click while browsing the Internet. Therefore, the profiles of these two protocols may become similar. In addition, some Streaming traffic was classified erroneously as Web-Browsing traffic, which is in accordance with a previous analysis presented in this section. This can be explained by the fact that, nowadays, most web pages

have embedded streaming contents and consequently, some of the data sets identified as Web-Browsing incorporate some characteristics of Streaming traffic. Therefore, the profile of some Streaming traffic may become similar to the one of Web-Browsing traffic (and vice-versa).

Table 5.6: Percentage of correctly classified *data-streams* using a multidimensional Gaussian distribution.

	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**100%**	0%	0%	0%	0%
CI	100%-100%	0%-0%	0%-0%	0%-0%	0%-0%
Snapshots	0%	**97.23%**	2.77%	0%	0%
CI	0%-0%	96.84%-97.63%	2.37%-3.16%	0%-0%	0%-0%
Web-Browsing	0%	0%	**97.20%**	1.83%	0.97%
CI	0%-0%	0%-0%	96.63%-97.77%	1.52%-2.15%	0.53%-1.40%
Streaming	0%	0%	8.13%	**91.50%**	0.37%
CI	0%-0%	0%-0%	7.12%-9.15%	90.45%-92.55%	0.19%-0.54%
Torrent	0%	0%	1.17%	3.13%	**95.70%**
CI	0%-0%	0%-0%	0.47%-1.86%	2.56%-3.71%	94.76%-96.64%

Table 5.7: Percentage of correctly classified *data-streams* using a multidimensional generic distribution.

	Identified as				
	NMap	**Snapshot**	**Web-Browsing**	**Streaming**	**Torrent**
NMap	**99.87%**	0%	0.13%	0%	0%
CI	99.71%-100%	0%-0%	0%-0.29%	0%-0%	0%-0%
Snapshots	0%	**97.94%**	2%	0%	0.06%
CI	0%-0%	97.48%-98.38%	1.55%-2.45%	0%-0%	0%-0.17%
Web-Browsing	0%	0%	**99.93%**	0.07%	0%
CI	0%-0%	0%-0%	99.82-100	0%-0.18%	0%-0%
Streaming	0%	0%	2.80%	**97.13%**	0.07%
CI	0%-0%	0%-0%	1.72%-3.88%	96.06%-98.20%	0%-0.18%
Torrent	0%	0%	6%	0.10%	**93.90%**
CI	0%-0%	0%-0%	4.51%-7.49%	0%-0.19%	92.44%-95.43%

The results obtained when modeling the distributions generated by the estimators with multidimensional generic approaches are shown in table 5.7. It can be seen that the classification results are very accurate, with an accuracy for all applications of more than 90%. This proves that multivariate generic approaches are suitable for an accurate legitimate traffic discrimination as well as to an accurate identification of low-impact illicit traffic.

Finally, table 5.4.2 shows the classification results when considering only two groups of applications: the legitimate and the illegitimate groups. It can be seen that the classification accuracy is very high when assigning *data-streams* to these mentioned groups. This

assesses that the proposed approaches are suitable for the identification of hosts running illicit applications.

Table 5.8: Percentage of correctly classified *data-streams* using multidimensional generic and Gaussian distributions.

	Generic distribution methodology Identified as		Gaussian distribution methodology Identified as	
	Licit traffic	**Illicit traffic**	**Licit traffic**	**Illicit traffic**
Licit traffic	**100%**	0	**100%**	0
CI	100%-100%	0-0	100%-100%	0-0
Illicit traffic	1.06%	**98.94%**	0	**100%**
CI	0.78%-1.34%	100%-100%	0-0	100%-100%

When comparing the classification results obtained in this section with the ones obtained with clustering algorithms, the accuracy are similar. However, the traffic traces used for analysis in this section are shorter since they comprise only 5 minutes of traffic. It can be concluded that the probabilistic approaches enable an accurate traffic mapping and an accurate identification of stealth and low-impact attacks using shorter traffic traces. This constitutes an important enhancement to the clustering approaches since these require longer traffic traces.

5.5 Conclusions

In this chapter, several identification methodologies were proposed. They rely on a multi-scale analysis of sampled traffic flows, enabling the identification of illicit activities on encrypted communications scenarios. The first proposed methodology consisted in modeling the distributions generated by the multi-scale decomposition estimators of the known data-streams using unidimensional probabilistic approaches. We started by analyzing only the first decomposition scales in an attempt of evaluating the presence of high-frequency events in the captured traffic. The used traffic *streams* had 5 and 15 minutes long, which enabled us to study the effect of the amount of data available and extracted from the traffic traces.

In order to optimize the classification approaches, our subsequent work focused on the use of all the available decomposition scales and more moments of analysis. Traffic classification was then based on identifying the time-scales where the different multi-scale estimators of the several Internet applications are better discriminated. Each application is conveniently identified based on the time scale where its multi-scale decomposition estimators are better separated.

The first proposed classification methodologies are based on two different approaches. The first uses unidimensional probabilistic modeling approaches. Using these probabilistic approaches, two classification methodologies were proposed: the first assumes that the

multi-scale decomposition estimators follow Gaussian Distributions and the other uses generic probabilistic models which compute the distances between the quantiles of the empirical distributions for classifying estimators extracted from unknown data-streams.

The second approach consisted in deploying multidimensional probabilistic models for traffic classification. A dimension was mapped to each of the chosen decomposition scales in order to generate a multidimensional space. In this manner, we can analyze the correlation between the estimators of the several decomposition scales (corresponding to dimensions) which allows us to infer more accurate distributions, modeling more accurately the distributions of the several Internet applications. Two models were then also used: the first deploys multidimensional Gaussian approaches while the second uses multidimensional generic approaches.

In order to evaluate the accuracy of the proposed classification methodologies, these were applied to some of the most used licit Internet applications and two popular illicit applications, and the results obtained show that they were able to accurately classify Internet traffic and identify illicit activities. Moreover, one of the methodologies was able to identify all illicit tested traffic and consequently, enable the identification of the network elements that are responsible for generating anomalous activities even in encrypted traffic scenarios.

By comparing the results presented in sections 5.4.2 and 5.4.1, we can conclude the use of multidimensional probabilistic approaches provides more accurate classification results. Indeed, the percentage of correctly classified data-streams of each studied Internet application increases when using such modeling approach. This is due to the fact that, such approaches, by generating a multidimensional space, allow the analysis of the correlation of the estimators of each analyzed dimension. This allows us to infer more accurate probabilistic approaches which enable a more accurate modeling of the different frequency components present in the analyzed *data-streams*. Therefore, the inferred distributions are more precise, allowing also a more accurate differentiation of unknown traffic. In the case of Multivariate Gaussian Distributions, it is assumed that the estimators follow Gaussian distributions in each one of the dimensions which, as shown in the tests verifying the suitability of this approach, constitutes a valid approach. On the other hand, for some applications, by not assuming Gaussian distributions, generic distributions can be deployed which may in fact increase the accuracy of the traffic mapping. As a drawback, multidimensional probabilistic modeling is a more intensive and resource consuming task.

Chapter 6

Enhancing Classification Approaches

6.1 Introduction

All the classification approaches presented in previous chapters enabled an accurate classification of the traffic generated by the most significant legitimate Internet applications, together with an accurate identification of the traffic generated by stealth and low impact Internet attacks. However, a considerable number of traffic samples extracted from each *data-stream* had to be analyzed in order to achieve such an accurate traffic mapping. Traffic profile changes were not analyzed since *data-streams* were considered to be generated by only one Internet application. Besides, we also believe that the detection of such profile changes will enable the detection of traffic presenting suspicious patterns only in some time intervals.

So, in order to be able to detect these profile changes, this chapter proposes an enhancement to the previously presented traffic classification methodologies : analyze and classify a *data-stream* using several classification windows of constant size. This will allow a more accurate traffic classification since each captured *data-stream* can then be assigned to one application according to the classification of the different windows. The accuracy increases with the increase on the number of analysis windows. This approach allows the identification of changes in the profile of the traffic generated by an Internet application as well as the identification of traffic presenting illicit patterns only in some periods of time in order to bypass detection and protection tools, as illustrated in figure 6.1. The use of appropriate threshold values enables the assignment of such traffic to different categories that can be used later by network managers to take appropriate counter-measures.

This chapter presents two windowed-based classification methodologies. Subsequently, some classification scenarios and the obtained results are presented and discussed. Finally, some conclusions are provided debating the advantages of the proposed improvement.

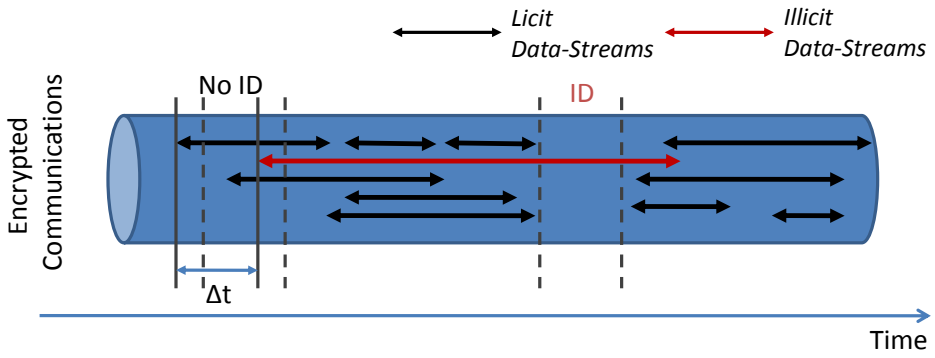

Figure 6.1: Window-Based Classification Concept.

6.2 Some preliminary definitions

Let us extend some definitions presented in section 3.3.4 in order to cope with classification enhancement proposed in this chapter. Let $w = 1, \ldots, W$ represent the w-th classification window, where W represents the total number of classification windows per *data-stream*. In addition, let

$$E^{w}_{a,\Gamma_z} = \{e^{i,w}_{a,\Gamma_z}, i = 1, \ldots, N; w = 1, \ldots, W\} \tag{6.1}$$

represent the estimators, as defined in (3.12), obtained from the w-th window of analysis for element Γ_z that indexes a scale $j, j = 1, \ldots, J$, at the order $q, q = 1, \ldots, Q$, moment of the wavelet decomposition estimators, of a stochastic process $s, s = 1, \ldots, S$, extracted from a *known data-stream* $i, i = 1 \ldots, N$, of an application a. The mean of the different estimators obtained from the multi-scale decomposition of *know data-streams* generated by an Internet application a, at each scale j, can be defined as:

$$\bar{e}^{w}_{a,\Gamma_z} = \frac{1}{N} \sum_{i=1}^{N} e^{i,w}_{a,\Gamma_z} \tag{6.2}$$

Moreover, let us define, for each application a and classification window w, a $L-$vector of mean values $\vec{e^{w}_{a}} = (\bar{e}^{w}_{a,\Gamma_z}), \forall l = 1, \ldots, L$,

On the other hand, let us define

$$U^{w}_{\Gamma_z} = \{u^{i,w}_{\Gamma_z}, i = 1, \ldots, M\} \tag{6.3}$$

as the estimators obtained for the w-th classification window for element Γ_z that indexes a scale $j, j = 1, \ldots, J$, at the order $q, q = 1, \ldots, Q$, moment as defined in (3.12), extracted from a stochastic process $s, s = 1, \ldots, S$, of an *unknown data-stream* $i, i = 1 \ldots, M$.

Each classification window, of each *data-stream* i, will be associated to an Internet application. The methodologies used to perform such association will be presented in the following

sections.

6.3 Window-Based Classification Approaches

In the following sub-sections, two different window-based classification approaches are presented, including multivariate Gaussian and generic probabilistic distributions to associate the multi-scale estimators obtained from the multi-scale decomposition of the different classification windows of each *unknown data-stream*.

6.3.1 Gaussian Window-Based Multidimensional Classification Approach

In this methodology, the distributions whose parameters are inferred from the estimators obtained from the multi-scale decomposition of the *known data-streams* of each studied application are modeled using multivariate Gaussian probabilistic approaches. Therefore, for the w-th classification window of an *unknown data-stream* i, the obtained estimators will be classified according to the probability of belonging to the multidimensional Gaussian probabilistic distributions of each Internet application, which is computed as:

$$P_{i,a}^{w} = \frac{1}{(2\pi)^{L/2}|\Sigma_a^w|^{1/2}} e^{-\frac{1}{2}(u_{\Gamma_z}^{i,w} - \vec{e}_a^w)^T \Sigma_a^{w\,-1} (u_{\Gamma_z}^{i,w} - \vec{e}_a^w)} \tag{6.4}$$

$\forall a = 1, \ldots, A, \forall i = 1, \ldots, M, \forall w = 1, \ldots, W$, where $\vec{e}_a^w$ is a L-vector, defined in (5.15), and $|\Sigma_a|$ is the determinant of Σ_a^w, which is the $L \times L$ covariance matrix of application a in window w.

The estimators obtained from the w-th classification window extracted from an *unknown data-stream* i will be assigned to the distribution of application α that maximizes the computed probability:

$$c_i^w = \alpha : P_{i,\alpha}^w = \max_a [P_{i,a}^w] \tag{6.5}$$

where c_i^w represents the classifier of the w-th classification window of the i-th *unknown data-stream*.

6.3.2 Generic Window-Based Multidimensional Classification Approach

Let us now assume that the distributions whose parameters are inferred from the estimators obtained from the multi-scale decomposition of the *known data-streams* of each Internet application follow a generic multidimensional distribution. Thus, for the w-th classification of the i-th *unknown data-stream*, the obtained decomposition estimators will be classified according to the distance $D_{i,a}^w, a = 1, \ldots, A$, to the distribution of application a:

$$D_{i,a}^w = \sqrt{(u_{\Gamma_z}^{i,w} - \overline{e}_{a,\Gamma_z}^w)^T \Sigma_a^{w\,-1} (u_{\Gamma_z}^{i,w} - \overline{e}_{a,\Gamma_z}^w)}, \forall a \tag{6.6}$$

where Σ_a^w is the covariance matrix.

The w-th classification window extracted from the unknown *data-stream* i will then be associated to the distribution of application α that minimizes the distance defined in (6.6), *i.e*:

$$c_i^w = \alpha : D_{i,\alpha}^w = \min_a \left[D_{i,a}^w \right], a = 1, \ldots, A \tag{6.7}$$

where c_i^w represents the classifier of the w-th classification window of the i-th *unknown data-stream*.

6.3.3 Data-Stream Classification

A final classification must be performed based on the several classifications that were obtained for the different classification windows of each analyzed *data-stream*. A *data-stream* can be classified as:

- Contains only traffic from an application α (or application group α),
- Contains a mixture of traffic from an application α (or application group α) and others,
- Does not contain traffic from application α (or application group α).

Therefore, let us define $N_{i,\alpha}$ as the weight of application α (or application group α) in *data-stream* i over W analyzed classification windows:

$$N_{i,\alpha} = \frac{1}{W} \sum_{w=1}^{W} \sum_{a=1}^{A} (c_i^w == a), \alpha = 1, \ldots, A \tag{6.8}$$

where A represents the total number of applications (or application groups) and operator == represents a comparison function which outputs 1 if both terms are equal and 0 otherwise.

The final classification of *data-stream* i is performed based on (empirically) predefined thresholds ϱ^- and ϱ^+:

$$\mathcal{C}_i = \begin{cases} \text{Application/Group } \alpha & \text{if } N_{i,\alpha} \geq \varrho^+ \\ \text{Mixture} & \text{if } \varrho^- < N_{i,\alpha} < \varrho^+ \\ \text{Not Application/Group } \alpha & \text{if } N_{i,\alpha} \leq \varrho^- \end{cases} \tag{6.9}$$

6.4 Results

This section presents the classification results obtained when enhancing the proposed classification methodologies with the use of several classification windows. Two classification scenarios are considered for analysis. The first uses a Gaussian window-based multidimensional methodology applied to non-sampled traffic metrics, the Inter-Arrival Time (IAT) and

the packet length, enabling a faster (*pseudo* real-time) classification. The second scenario uses a generic window-based multidimensional classification approach to identify illicit traffic or traffic composed by mixtures of licit and illicit excerpts.

6.4.1 Gaussian Window-Based Multidimensional Classification Based on Non-Sampled Traffic Metrics

In this section, we present the classification results obtained when using the Gaussian window-based multidimensional methodology applied to non-sampled traffic metrics (Inter-Arrival Time (IAT) and packet length). In this scenario, we included one additional application, P2P-TV, whose traffic was captured as described in [Pet10]. A *pseudo* real-time traffic classification paradigm was achieved since classification was faster, while maintaining a similar accuracy level. The IAT ($S = 1$) and length ($S = 2$) of each packet in the download direction and IAT ($S = 3$) and length ($S = 4$) of each packet in the upload direction were the chosen stochastic traffic processes. In order to evaluate the accuracy of the proposed classification approach and its dependency on the number of the classification windows, 100 independent simulations were performed. For each one, 20 *data-streams* ($N = 20$) of each Internet application were chosen as *known data-streams*. The remaining *streams* ($M = 30$) were used as *unknown data-streams* and were further classified. The classification approach presented in section 6.3.1 was used for classifying each one of the different classification windows based on the distributions of the different studied applications. The parameters of such distributions were inferred from the estimators obtained from the multi-scale decomposition of *known data-streams* of each Internet application. The size of each window was then set to 128 packets. The chosen thresholds were $\varrho^+ = 90\%$ and $\varrho^- = 10\%$. The use of several classification windows allows an increase on the classification accuracy, since more differentiating characteristics can be inferred from the traffic of the different studied Internet applications. Each *data-stream* was assigned to an Internet application according to (6.8) and (6.9).

Let us start by analyzing the time, shown in table 6.1, that is required to obtain the number of packets necessary for performing an analysis with the different number of analysis/classification windows. For web-browsing traffic, it can be seen that the time that is required in order to have enough metrics for one analysis/classification window is 14 seconds, for the upload traffic, while only 11.3 seconds are required for the download traffic. The amount of time for the download direction is lower than the one of the opposite direction, because more traffic flows down to the user due to the download of the requested web-pages. However, for web-browsing applications, these values depend on the user profile and on the number of performed requests. For the BitTorrent traffic, only 3.35 and 2.69 seconds are necessary to obtain a sufficient number of packets for one analysis window, for the download and upload directions respectively. The amount of traffic generated by peers when downloading, and simultaneously uploading, a file allows us to capture the required number of packets in a

Table 6.1: Time, in seconds, required for traffic classification

	Number of Analysis Windows			
	1		2	
	Upload	**Download**	**Upload**	**Download**
Web-Browsing	14.14	11.38	15.20	12.28
Torrent	3.35	2.69	3.61	2.90
P2P-TV	0.81	0.78	0.87	0.84
Streaming	6.99	6.63	7.53	7.13
NMap	7.78	6.11	8.40	6.57
Snapshots	85.98	51.12	92.20	53.33
	Number of Analysis Windows			
	3		4	
	Upload	**Download**	**Upload**	**Download**
Web-Browsing	16.26	13.18	17.35	14.09
Torrent	3.87	3.11	4,13	3.32
P2P-TV	0.93	0.93	0.99	0.96
Streaming	8.06	7.63	8.60	8.13
NMap	9.03	7.03	9.66	7.50
Snapshots	98.80	56.92	105.37	60.53

small period of time. For performing an analysis with 4 classification windows, these values increase to 4.13 and 3.32 seconds, for the download and upload directions respectively. When classifying P2P-TV traffic, these values decrease considerably due to the fact this application generates a significant amount of traffic, with very low IATs, which is caused by the download and upload of a TV channel broadcast using P2P networks. The amount of time necessary for obtaining a sufficient number of packets for performing an analysis with 4 classification windows does not reach 1 second, which confirms that traffic generated by this application can be quickly identified. About 8 seconds are required to perform classification using four analysis windows for the video Streaming traffic, because this application generates a less amount of traffic when compared with P2P file download and P2PTV applications. In fact, Video-Streaming consists of the transmission of a video between a server and a client, while for P2P-TV many peers are involved in the video transfer and, consequently, more traffic is generated. About 9 seconds of upload traffic and 7 seconds of download traffic are required for capturing a sufficient number of packets for performing a classification based on 4 analysis windows for the NMap application. For the Snapshots traffic, these values increase considerably due to the fact that uploads of stolen confidential information are only performed when users perform requests on a browser. In addition, the number of exchanged packets per upload is not significant since only small images or text files are sent to a remote server.

The classification accuracy, together with their 95% confidence intervals, as a function of the number of used classification windows, is illustrated in figure 6.2. It can be seen that for one classification window the results are already quite accurate, being the Snapshot and

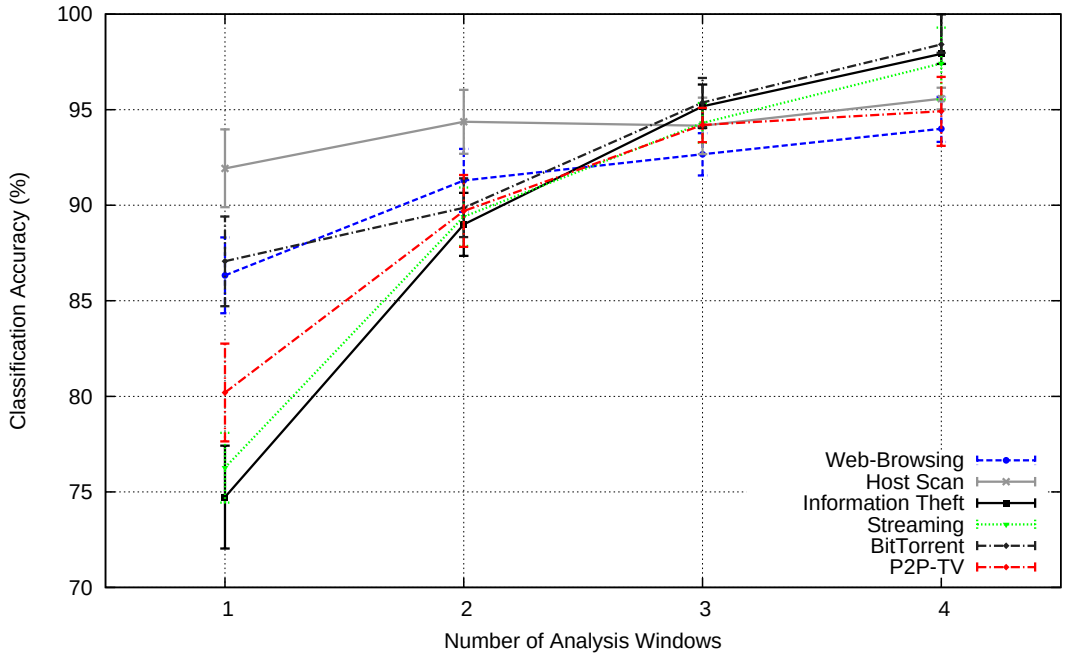

Figure 6.2: Classification Accuracy.

Video Streaming applications the ones that present the lowest accuracy values. However, by increasing the number of classification windows, the accuracy increases for all studied Internet applications, as expected. In fact, when using 4 classification windows the accuracy is higher than 90% for all applications. This is due to the fact that more metrics can be obtained from the captured *streams* when using more analysis windows, which enables the use of more accurate probabilistic distributions leading to a better discrimination between the traffic generated by the different applications.

6.4.2 Identification of Illicit Traffic using Generic Window-Based Multidimensional Classification

Now, the considered stochastic processes are the byte count per sampling interval (0.1 seconds) in the download ($s = 1$) and upload directions ($s = 2$) and the packet count per sampling interval (0.1 seconds) in the download ($s = 3$) and upload directions ($s = 4$). These estimators were computed for the first eight time-scales ($J = 8$) of the first three order moments ($Q = 3$). The width, Δt, of the time-windows over which *data-streams* were sampled was approximately equal to 100 seconds, which at a sampling interval of 0.1 seconds gave us 1024 samples per *data-stream*, per sampling window. We considered 8 classification windows per *data-stream*.

From all captured *data-streams*, 20 of them from each application were randomly chosen as *known data-streams* ($N = 20$) and some samples were extracted from these streams. The

Table 6.2: Percentage of correctly identified *data-streams* of mixed traffic.

	Identified as		
	Licit	**Illicit + Licit**	**Illicit**
Licit	**100**%	0%	0%
Illicit	3%	**97**%	0%
Licit + Illicit	10%	0%	**90**%

remaining ones were used as *unknown data-streams* ($M = 30$) and samples were also extracted from them and classified in order to evaluate the efficiency of the classification methodology. The approach presented in section 6.3.2 was used for associating each one of the classification windows to the corresponding Internet application. Each *data-stream* was assigned to an application group according to (6.8), considering the following two application groups:

- Licit traffic: HTTP (Web-browsing), Streaming and Torrent,
- Illicit traffic: NMAP and Snapshots.

The classification was performed as described in section 6.3.3 and with thresholds $\varrho^+ = 90\%$ and $\varrho^- = 10\%$. The obtained results were very accurate, as shown in Table 6.4.2. The identification of embedded illicit patterns in legitimate communications was achieved for 90% of all traffic *streams* that had hidden illicit patterns. These results confirm that the proposed enhancement and the use of appropriate thresholds allow an accurate identification of hidden illicit patterns in legitimate traffic *streams*.

6.5 Conclusions

Methodologies that can accurately and timely classify Internet traffic and Internet attacks are critical in order to assure several network management tasks. In fact, network resources can be optimized, better QoS can be achieved and security can also be improved by accurately identifying traffic with suspicious behaviors.

This chapter presented an enhancement to all classification methodologies that were presented in chapter 5. By analyzing, decomposing and classifying *data-streams* over several classification windows, the classification accuracy was increased, as well as the identification of profile changes and of hidden illicit patterns embedded in licit traffic. The ability to identify these behaviors was achieved by using several sampling windows over which different traffic metrics were extracted and decomposed. This makes our classification approaches suitable for the detection of some of the most stealth and well engineered attacks and intrusion attempts. In addition, the timely classification of Internet traffic and of illicit patterns was enabled by using non-sampled traffic metrics, such as the IAT and packet length, for each packet of a *data-stream*. In this manner, we were able to build a *pseudo* real-time profile for

each *data-stream* that is updated with the capture of new packets. The obtained results were sufficiently accurate and also proved that the proposed enhancement is suitable for a real-time traffic classification paradigm. In addition, two commonly deployed security attacks were also accurately and timely classified, which proved that the proposed approaches are suitable for security attacks and intrusion detection.

Chapter 7

User Profiling for Network Management Purposes based on Traffic Scalograms

7.1 Introduction

This chapter proposes and describes an approach for accurate profiling of the users connected in wired/wireless networks. As already discussed, user profiling is a critical task for many network management tasks. In fact, the ability to accurately build efficient user-profiles can have a crucial importance for many different aspects. To begin with, one can more easily infer the bandwidth and delay requirements that are more suitable for a certain user and network resources can then be optimized and better distributed among several users. Therefore, better Quality-of-Service (QoS) standards can be achieved for every connected client. Besides, by accurately profiling connected users, network managers can create groups of users requesting similar contents, which eases the delivery of appropriate and related contents and services. In this way, more accurate business models can be built, leading to increasing revenues. Security can also be effectively improved since it is possible to detect users presenting illicit profiles or profiles presenting unknown applications, triggering alarms and providing counter-actions, such as disconnecting malicious users. In can be concluded that an accurate user profiling is crucial for allowing all the connected clients to experience a better QoS and allowing network managers to perform a better management of the network infrastructures and resources.

There are several definitions for an user profile [GA05], but a common definition can state that it consists of a description of the user's interests, behaviors and preferences. Therefore, the process of creating such profiles can be seen as the process of gathering the appropriate information in order to obtain all these characteristics. In this work, we adopt a very specific

definition of user-profile, which is more oriented to the set of web-applications that each user runs and interacts with. Consequently, the focus is placed on applications that allow users to share on-line information and contents. Our analysis is achieved through a promiscuous wireless monitoring approach in which monitoring probes, which do not require authentication with the Access Point (AP), are user for promiscuously monitoring all connected clients and for collecting different layer 2 traffic metrics. We then perform a wavelet decomposition at different scales of analysis as described in section 3.3.3. By decomposing captured Internet traffic generated by different clients running different web-based applications and analyzing it at the different scales, we can build a *Multi-Scale Application Signature* that depicts the different frequency components characteristic of each studied on-line web-based applications. As will be shown later, these applications require different user interactions, thus creating different traffic patterns that lead to distinct frequency profiles. It is then possible to map all these components into the corresponding user and/or network event and to accurately assign the captured traffic into its originating on-line web-based application. After inferring these characteristic signatures, classification can be performed as quickly as a perfect match is obtained. The speed of classification depends on the profile characteristics and can range from few seconds to few minutes, depending if differentiating characteristics appear at network/service scales or human scales, respectively.

Our profiling approach is suitable for being deploying in a user profiling module as the one described in section 1.3. Such module can deploy the approach presented in this chapter for associating the analyzed traffic with the corresponding Internet application and for building accurate user profiles. Such task should be an off-line methodology more suitable to be performed using CWTs due to the computational complexity of this transform. as discussed in section 3.3.2. In addition, our definition of user-profile requires more frequency details for building accurate frequency descriptions of the traffic generated by the different web-applications and CWTs are a more appropriate tool for such detailed analysis.

The proposed profiling approach will be validated by analyzing traffic sent to several clients connected to a 802.11 wireless network and inferring the applications that are being run by the different clients. The obtained results prove that it is possible to accurately assign traffic to its originating on-line web-application, thus providing a reliable and accurate description of the usage of web-based applications. The use of Layer 2 metrics allows our classification approach to become appropriate for the classification of encrypted traffic, where the payloads of the packets are not available, and also to circumvent possible technical, legal and privacy restrictions that prevent the inspection of the contents of the packets.

The following sections will present our profiling methodology together with some classification results obtained when considering two different scenarios. The first assumes only that legitimate applications are used by the connected clients and, consequently, no illicit traffic is considered for analysis. In such scenario, we wanted to evaluate the ability of our profiling ap-

proach of accurately differentiating the traffic generated by licit and allowed on-line Internet applications. The second scenario assumes compromised hosts in the monitored network and two stealth and low-impact attacks were emulated for assessing the ability of our approach of identifying such traffic. In addition, we also performed a discrimination between some important legitimate on-line Internet applications.

7.2 Classification Methodology

In this section, we present a simple classification approach that allows us to illustrate the ability to accurately classify traffic based on the analysis proposed in section 3.3. The main concept consists of dividing the frequency spectrum into different regions and evaluating the power of the CWT decomposition multi-scale estimators in those regions. Three different frequency spectrum regions, together with their corresponding events, were shown in Figure 3.1. As explained in section 3.3, low frequency components account for human events that, in the Internet world, are associated to human/user behaviors and actions. Between the low and high frequency regions, we have created a mid-range frequency region that accounts for network events such as the creation of traffic sessions and the corresponding traffic control mechanisms. Finally, in the high-frequency spectrum region, protocol and Internet events such as packets arrivals are accounted for. All these mappings into events, which can then be associated to the corresponding Internet applications, allow a simple but effective traffic assignment. In this manner, we can assess and quantify the different network and user mechanisms and the interactions present in the traffic of each one of the mentioned Internet applications.

By defining characteristic regions of the scalogram statistics, for the different applications, in different frequency sub-sets, it is possible to identify profiles presenting components characteristic to each web-application. Such regions are inferred from the scalograms obtained from the decomposition of the *known traffic* of each web-application. Let us consider the (positive) region R_a^+ as the region defined as a function of a frequencies (positive) sub-set $\mathbf{s}_a^+$ and energy variation (positive) sub-set $\mathbf{\Sigma}_a^+$ for which we always have the characteristic statistical values of application a. Moreover, we define the (negative) region R_a^- as a function of a frequencies (negative) sub-set $\mathbf{s}_a^-$ and energy variation (negative) sub-set $\mathbf{\Sigma}_a^-$ for which we never have characteristic statistical values of application a.

$$R_a^+ = f(\mathbf{s}_a^+, \mathbf{\Sigma}_a^+) \wedge R_a^- = f(\mathbf{s}_a^-, \mathbf{\Sigma}_a^-) \tag{7.1}$$

A traffic trace process $x(t)$ is classified as belonging to web-application a if, for all scales belonging to sub-set $\mathbf{s}_a^+$, the energy standard deviation $\sigma_{x,s}$ belongs to region R_a^+ and, simultaneously, for all scales belonging to sub-set $\mathbf{s}_a^-$ the energy standard deviation $\sigma_{x,s}$ does not belong to region R_a^-:

$$C(x) = a \Leftarrow \forall s \in \mathbf{s}_a^+, \sigma_{x,s} \in R_a^+ \wedge \forall s \in \mathbf{s}_a^-, \sigma_{x,s} \notin R_a^- \tag{7.2}$$

The classification decision can be made as soon as all conditions are met. Note that, even if time $\mathbf{T}$ grows and allow more classification precision, decisions can nevertheless be made with small $\mathbf{T}$ sub-sets (short-time analysis and decision).

The inference of regions R_a^+ and R_a^- (defined by $\mathbf{s}_a^+, \mathbf{\Sigma}_a^+, \mathbf{s}_a^-, \mathbf{\Sigma}_a^-$) can be performed by solving the following optimization problem:

$$\max_{\mathbf{s}_a^+, \mathbf{\Sigma}_a^+, \mathbf{s}_a^-, \mathbf{\Sigma}_a^-} \left(\sum_{\forall i \in \mathbf{I}_a} C(i) == a \right) \wedge \min_{\mathbf{s}_a^+, \mathbf{\Sigma}_a^+, \mathbf{s}_a^-, \mathbf{\Sigma}_a^-} \left(\sum_{\forall i \notin \mathbf{I}_a} C(i) == a \right), \forall a \tag{7.3}$$

where == represents a comparison function which outputs 1 if both terms are equal and 0 if terms are different. $\mathbf{I}_a$ represents the subset of processes (known as) belonging to web-application a. Within the scope of this chapter, this optimization problem was solved (not for the optimal solution) using exhaustive search. However, more advanced algorithms can be applied to find (sub)optimal solutions.

Several regions can then be created, in the different frequency sub-sets, for each studied web-application a. The higher the number of regions of an application, the higher the ability to analyze the different frequency components and consequently, a more accurate traffic mapping can be achieved. An algorithm was created to automatically define such regions while satisfying the presented conditions by using known simple geometrical equations, such as ellipses.

7.3 Results

In order to verify the accuracy of the presented approach, traffic from several connected clients using the studied Internet application classes was captured in the wireless network of University of Aveiro since we could guarantee the ground-truth of the different traffic traces. This traffic was captured by the monitoring probes in different points of the network and at different time moments. We inferred the number of captured packets per sampling interval (0.1 seconds) in the download direction, *i.e* traffic sent to the local hosts. The multi-scale analysis presented in section 3.3.2 and the analysis depicted in section 7.2 was applied to all captured data-streams. The mother wavelet used was the fourth derivative of the Gaussian function which is defined as follows:

$$\psi(t) = C \frac{d^4}{dt^4} e^{\frac{-t^2}{2}} \tag{7.4}$$

where C is such that $\int_{-\infty}^{+\infty} |\psi(t)|^2 dt = 1$.

The normalized energy ($\hat{E}_x(\tau, s)$) in all time slots ($\tau \in \mathbf{T}, \mathbf{T} = \{0, \ldots, 3000\}$), and time-scales 1 to 128 ($s \in \mathbf{S}, \mathbf{S} = \{1, \ldots, 128\}$), for all *data-streams*. The obtained scalograms were normalized for the whole length of the process, as described in equation 3.9 and the several differentiating regions were inferred according to equations (7.1) to (7.3).

A note should be made about the classification results and the corresponding confidences intervals that will be presented in the following sections. Few iterations were performed due to the low number of traffic traces that could be collected and such results, together with the corresponding confidence intervals, should then be analyzed as a proof-of-concept assessing the ability of the presented approach of depicting accurately the different frequency components and of performing an accurate traffic assignment.

7.3.1 Legitimate Internet applications

Let us first evaluate the ability of the proposed approach of discriminating the traffic generated by legitimate on-line services. For such purpose, five significant on-line Internet services were considered for this analysis: on-line news, on-line mail, social networking, photo sharing and video services. Several usage scenarios were created to generate traffic from these services: for example, on-line news traffic was generated by visiting the most important Portuguese newspaper site and browsing through the available news; on-line video download traffic was generated by watching videos in YouTube; in order to generate traffic from an on-line photo-sharing application, an account was created in Flickr and only the traffic generated while browsing other users' photos was considered for analysis; on-line e-mail traffic was generated by using the services offered by GMail, specifically traffic generated only by the automatic synchronizations between the client web-terminal and the GMail server; finally, social networking traffic was generated by using an account created on Facebook and interacting with the news updates coming from the remaining connected users, which does not include chatting and gaming. Table 7.3.1 shows the mapping between the available web-applications and the web-services that were used to generate traffic from each application.

Figures 7.1 to 7.5 show the captured traffic metrics, the download rate in bytes per second, sampled in 0.1 seconds intervals together with the corresponding wavelet scalograms for the different web-applications that were previously mentioned. The analysis of these figures reveals differentiating characteristics that are caused by the distinct traffic patterns presented by these applications, whose origin lies in the distinct human and network/service interaction characteristics. On-line news traffic (Figure 7.1), for example, presents several aperiodic peaks of short duration and considerable amplitude. Such peaks are caused by the user clicks on hyper links while browsing through the available news, causing the download of a new page that presents the requested news, thus creating considerable low frequency components. In addition, the scalograms generated by this application present some considerable mid-frequency components, due to the considerable number of created Transmission Control Protocol (TCP)

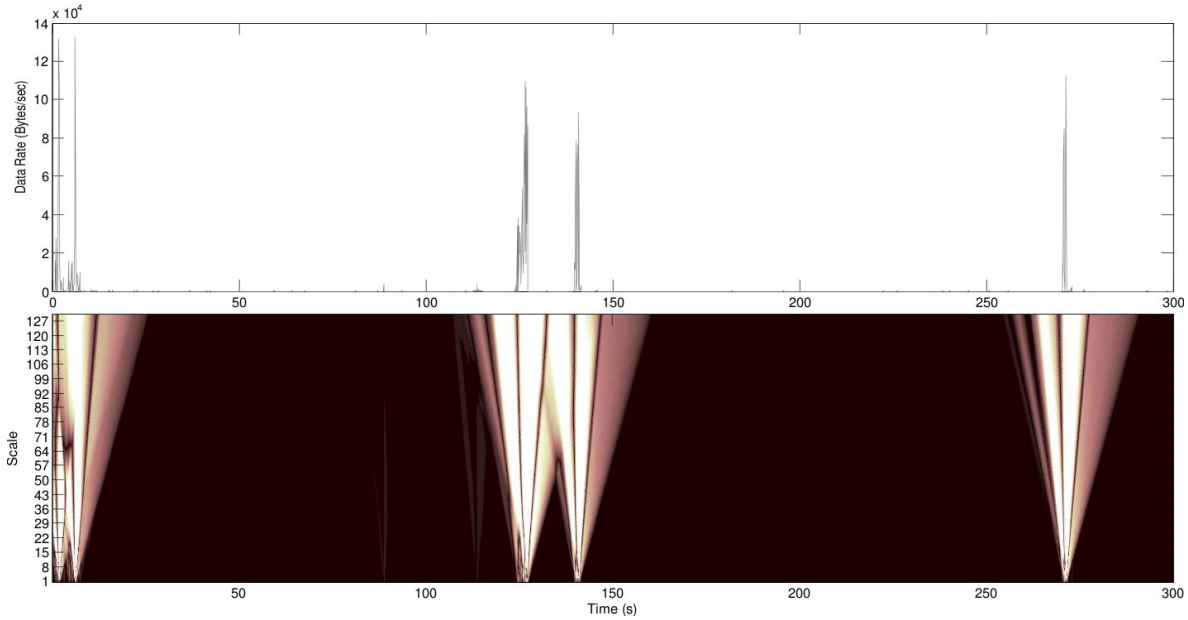

Figure 7.1: On-Line News Traffic Patterns and corresponding Wavelet Scalograms

sessions, while there are some considerable high frequency components due to packets arrivals. On-Line video services (Figure 7.2) generate high-bandwidth traffic with a low IAT between packets, which is caused by the download of the requested video at the full available network bandwidth. Consequently, there are considerable high-frequency components, caused by packets arrivals, while there are no considerable low-frequency components because the number of user clicks is not so relevant. On-line Photo-sharing (Figure 7.3) applications usually generate several traffic peaks with pseudo-periodicity, due to the pseudo-periodic clicks that are performed by the user while requesting for another picture. Such peaks are usually of low amplitude, since they only consist on the download of one picture using a single TCP session. Consequently, we can notice several high frequency components, of low amplitude, spread over the corresponding scalogram, while there are also some low frequency components. On-line email applications (Figure 7.4) generate traffic presenting very low frequent traffic peaks, corresponding to the initial and automatic synchronization between server and client. These peaks have very short duration and are less frequent than the ones of the previously presented on-line applications. Therefore, there are small high-frequency components caused by the synchronization traffic that merely checks for new e-mails, while low-frequency components are not widely spread over the traffic scalogram due to near periodical nature of network/service events. Finally, on-line social networking applications (Figure 7.5) generate traffic presenting more frequent traffic peaks, of lower amplitude, which are generated by the status updates created by other connected users, which usually consist only of text messages. Therefore, there are less low-frequency components, while the high-frequency components are also less present in the process due to the small amount of traffic exchanged.

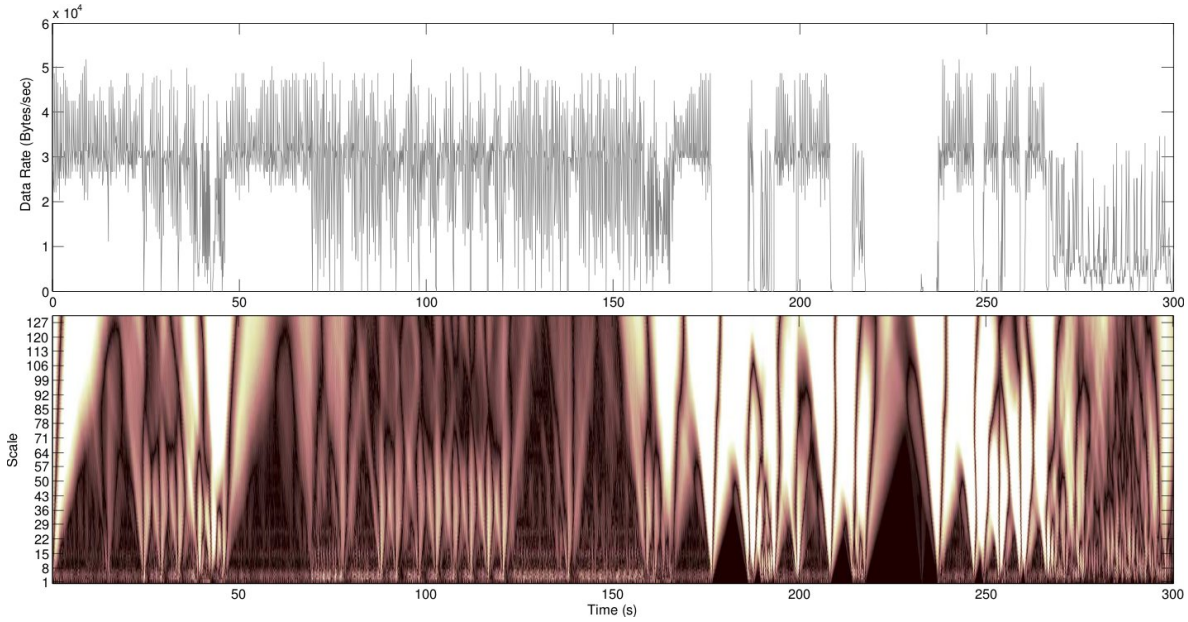

Figure 7.2: On-Line Video Traffic Patterns and corresponding Wavelet Scalograms

Figure 7.6 presents a graph of the standard deviation (over time) of the multi-scale decomposition estimators versus the corresponding frequency, or scale of analysis, of four different flows (randomly chosen from the data-set) belonging to each web-application. According to this figure, by analyzing the variation profile of the network process energy throughout the whole range of frequencies it is possible to obtain an accurate association between a given traffic flow and the application that originated it, simply by performing an analysis in the differentiating regions, as explained in section 7.2. The depicted regions were inferred by solving the minimization processes described in equations (7.1) to (7.3), using exhaustive search algorithms in predefined solution sets and including the complete dataset.

Let us begin by analyzing the inferred regions and describing the differentiating traffic characteristics that led to them, since each region characterizes a sub-frequency range that is mapped into specific human and network/service events. For instance, region A was assigned to on-line e-mail application since it comprises very-low frequency events, usually triggered by very rare events. For the mentioned application, such events are generated by the initial download of the e-mail web interface and the periodic synchronizations between the application running on the client side and the remote server. Region B was assigned on-line news, photo-sharing and social networking applications since it encompasses low frequency events. These include user clicks requesting new contents suitable to on-line news browsing or clicks performed when browsing through pictures provided from an on-line photo-sharing community or interactions generated by social networking applications and their news feeds. Therefore, the differentiation between these three applications will have to include more mid and high frequency regions. Regions D and E encompass mid-frequency events such as the

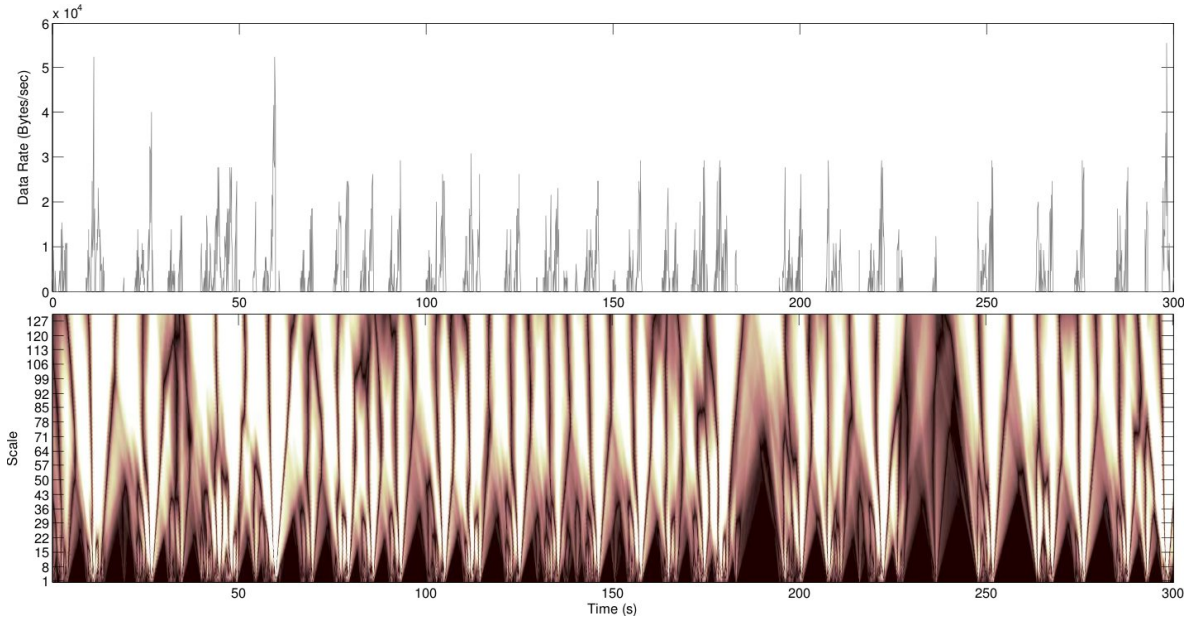

Figure 7.3: On-Line Photo Sharing Traffic Patterns and corresponding Wavelet Scalograms

ones associated with TCP and HTTP interactions. Region D was assigned to the on-line news application since it includes traffic presenting higher energy variation in the corresponding frequency range, implying that a higher number of TCP sessions are created. Such behavior is more likely to be created by user clicks on on-line news sites, since the download of a new page comprises several TCP and HTTP sessions. On the other hand, the second region (E) was assigned to social-networking applications as it comprises traffic presenting a lower number of created Internet sessions, since there is lower energy variation in that region of frequencies. This is more characteristic of social-networking applications, since the interaction with the news feed and the corresponding status updates create less TCP sessions than applications mapped into region D. Region C comprises traffic from the on-line video and photo-sharing applications since it includes traffic presenting low energy variation on low-frequency events, such as user clicks, or events with similar inter-event time. Both characteristics can be associated to on-line video applications, since they require a low number of user clicks, and photo-sharing applications, where the time between clicks presents lower variation. Region F comprises traffic generated by on-line news and video services and is characterized by a significant amount of high frequency events, such as packets arrivals, suitable to describe the high-frequency profile created by on-line video applications or web-pages with embedded video, characteristics of on-line news applications. On the other hand, region H can be seen as a region characteristic of applications such as photo-sharing which typically present a low number of events on this frequency range. Indeed, a low number of packets is required to download a shared picture. Region G is located between the two previously mentioned regions and presents more significant high-frequency components than region H

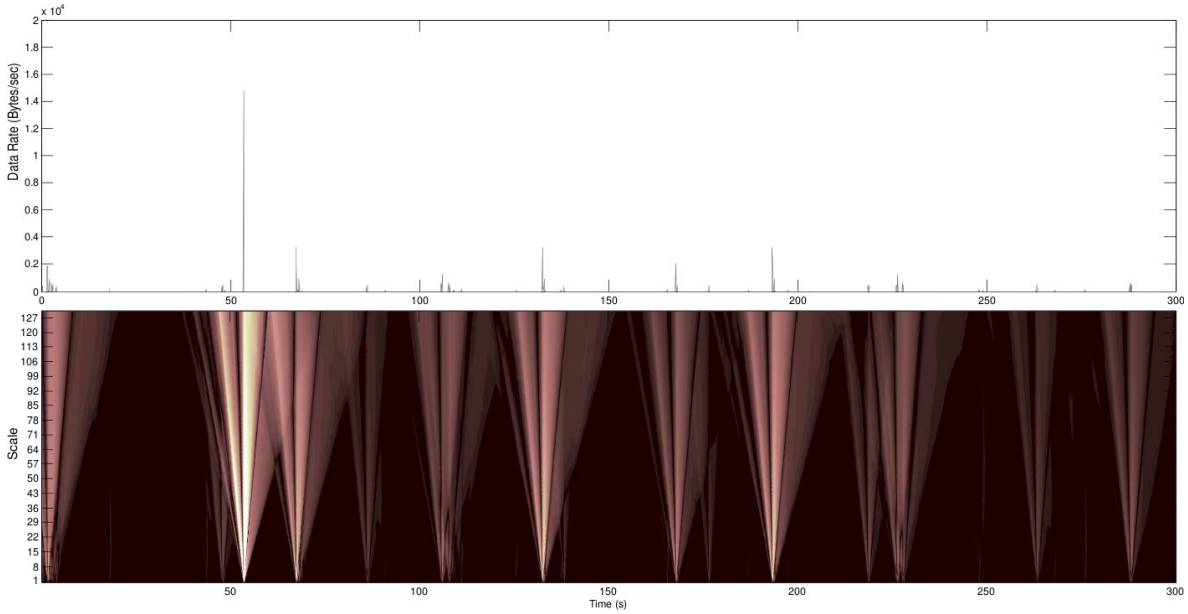

Figure 7.4: On-Line e-mail Traffic Patterns and corresponding Wavelet Scalograms

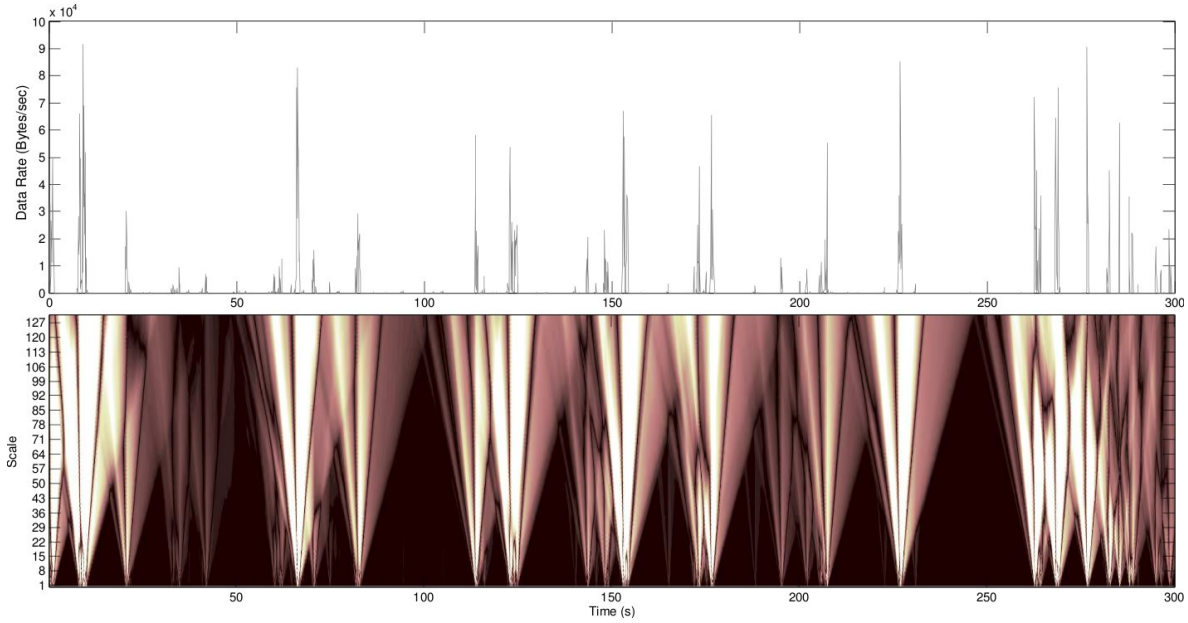

Figure 7.5: On-Line Social Networking Traffic Patterns and corresponding Wavelet Scalograms

and less high-frequency components than region F. Such region can be used to identify flows with a considerable (but not high) packet arrival rate or presenting a deviation from the normal profile of the generating application. Each studied web-application was mapped into one or more of the presented regions, as shown in table 7.3.1, and an algorithm was created to detect and classify the scalograms of the different captured traffic streams. Such algorithm

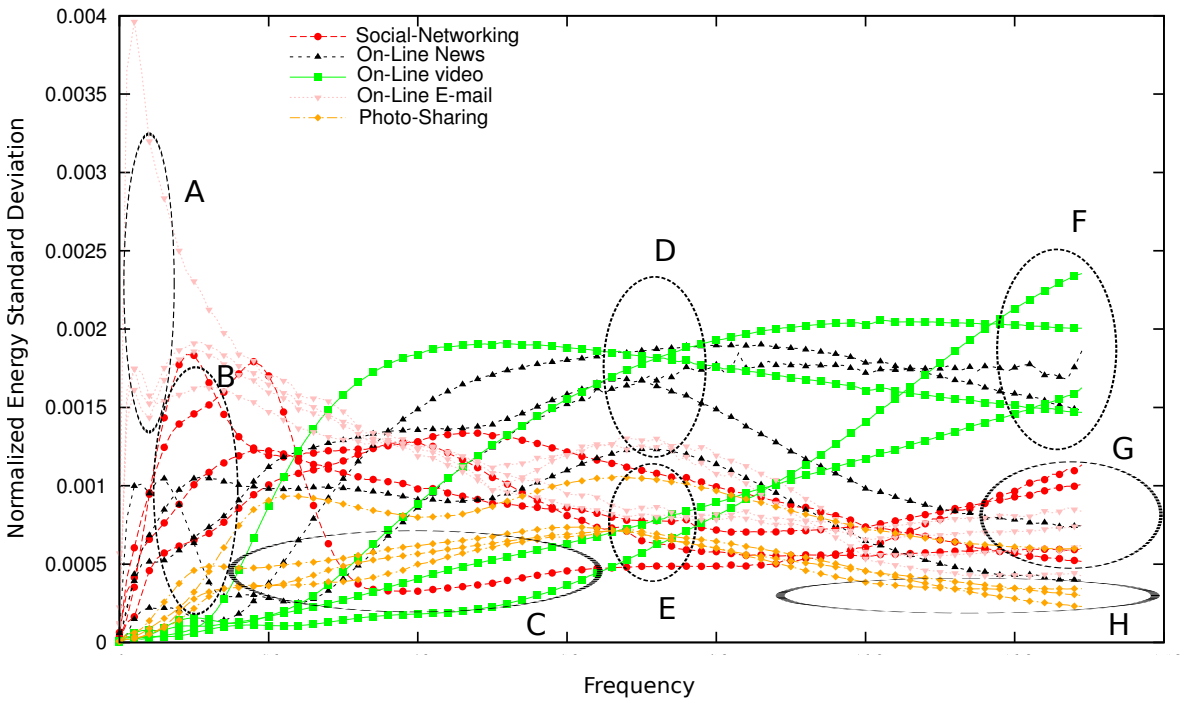

Figure 7.6: Differentiating Regions

simply needs to detect the variation of frequency components of the several scalograms in the inferred regions, mapped into web-applications as described in table 7.3.1, and assign the corresponding traffic accordingly.

Table 7.1: On-Line Applications with their corresponding web sites and frequency mapping regions.

Service	**Web site**
On-Line News	Publico (`www.publico.pt`)
On-Line Video	YouTube (`www.youtube.com`)
Photo Sharing	Flickr (`www.flickr.com`)
On-Line E-mail	GMail (`www.gmail.com`)
Social Networking	Facebook (`www.facebook.com`)

Let us now analyze the classification results that were achieved by applying the above presented approach, which are shown in table 7.2 together with their confidence intervals. Most of the generated traffic is accurately mapped into the corresponding web-application. However, there are some classification errors that need to be explained. The association of some on-line traffic to video services can be due to the fact that some requested news presented embedded videos. Therefore, the profile can become similar to the one corresponding to video applications. Some flows from web-video traffic were assigned to on-line news, which can happen when watching several small duration movies, since in this case the user can make more clicks in order to request for new contents, creating significant low-frequency compo-

Table 7.2: On-Line Applications with their corresponding frequency mapping regions and classification results.

Service	Regions	Classification Accuracy
On-Line News	B and D and (G or F)	**88.00%**
CI		87.30%-88.70%
On-Line Video	C and F and not(B)	**88.90%**
CI		88.00%-89.80%
Photo Sharing	B and E and H	**85.72%**
CI		84.32%-87.12%
On-Line E-mail	A and (E or D)	**87.50%**
CI		86.40%-88.60%
On-Line E-mail	B and E and G	**88.55%**
CI		87.30%-89.80%

nents that are characteristic of on-line web-applications. Some classification mistakes also occurred for photo-sharing applications where some flows were classified as social-networking flows, which can occur when a user receives some status updates from other users through the photo-sharing service. Some web e-mail traffic was also associated to social networking applications, which can be due to the fact that when there is a small amount of e-mail updates the application profile gets more similar to social networking small message exchange. Finally, some social networking flows were associated to on-line news, which can occur if a considerable number of status updates occurs in a small time frame.

7.3.2 Identification of illicit traffic

Let us now evaluate the accuracy of our approach for the identification of low-impact and stealth attacks. Therefore, the studied applications comprised two emulated security attacks to verify the ability of identifying compromised hosts and three other legitimate applications (On-Line News, On-Line Video and On-Line Photo-Sharing). The first emulated attack consists of an host-scan using the well-known application NMap [Lyo09] for replicating the behavior of a compromised host scanning for available services, and corresponding vulnerabilities, in other connected hosts. The second emulated illicit application was an Information Theft attack which consisted of taking snapshots of the users' desktops and uploading the captured pictures, every time the user performed a click, to a remote server in order to steal confidential information. Figures 7.7 and 7.8 show the the download rate, in bytes per second sampled in 0.1 seconds intervals, and the corresponding wavelet scalograms for the two considered illicit applications. These figures reveal the specificities of the emulated attacks. To begin with, host scan traffic presents small peaks corresponding to the response to the several Syn packets sent to verify if a port is open or not. On the other hand, Information Theft traffic presents non-periodic traffic peaks which correspond to the acknowledgments sent by the remote server when receiving the uploaded snapshots of the user's screen. Such upload are performed when the user performs a click on a web-page.

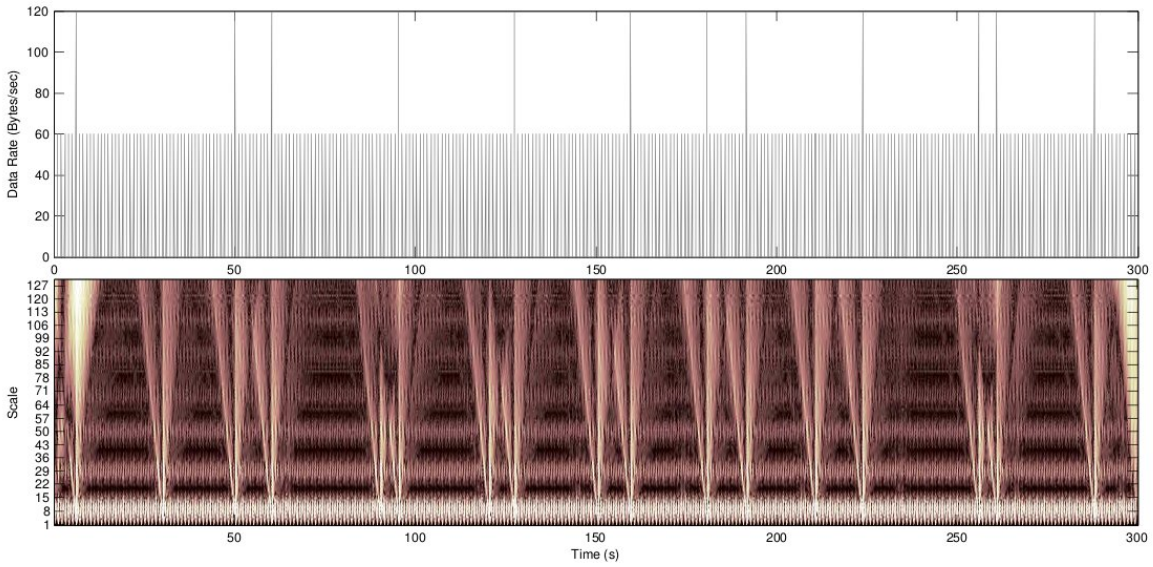

Figure 7.7: Host Scan Traffic Patterns and corresponding Wavelet Scalograms

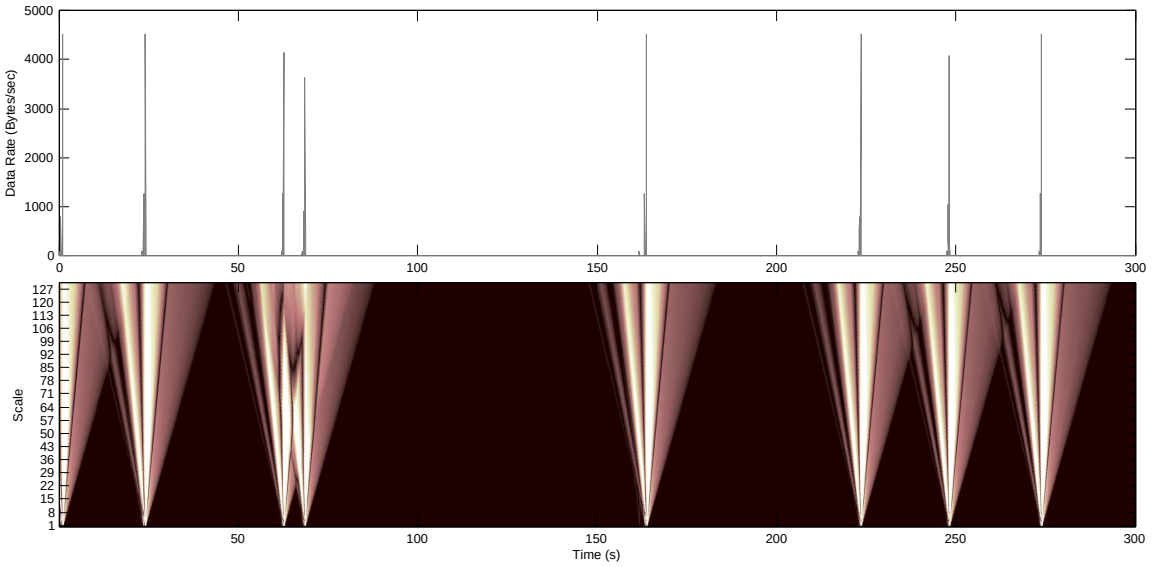

Figure 7.8: Information Theft Traffic Patterns and corresponding Wavelet Scalograms

Several differentiating regions, shown in Figure 7.9 and mapped into the corresponding application in table 5.4.1, emerged in the frequency spectrum. Region A was associated to illicit traffic and encompasses very low frequency events which are created by commands sent to compromised hosts in order to perform a scan or an upload of stolen confidential informations. This region can, thus, be associated to stealth attacks. Network scans also map to another differentiating region (region E), since these scans do not generate substantial mid range frequency components due to the low variance between scanning probes and to the

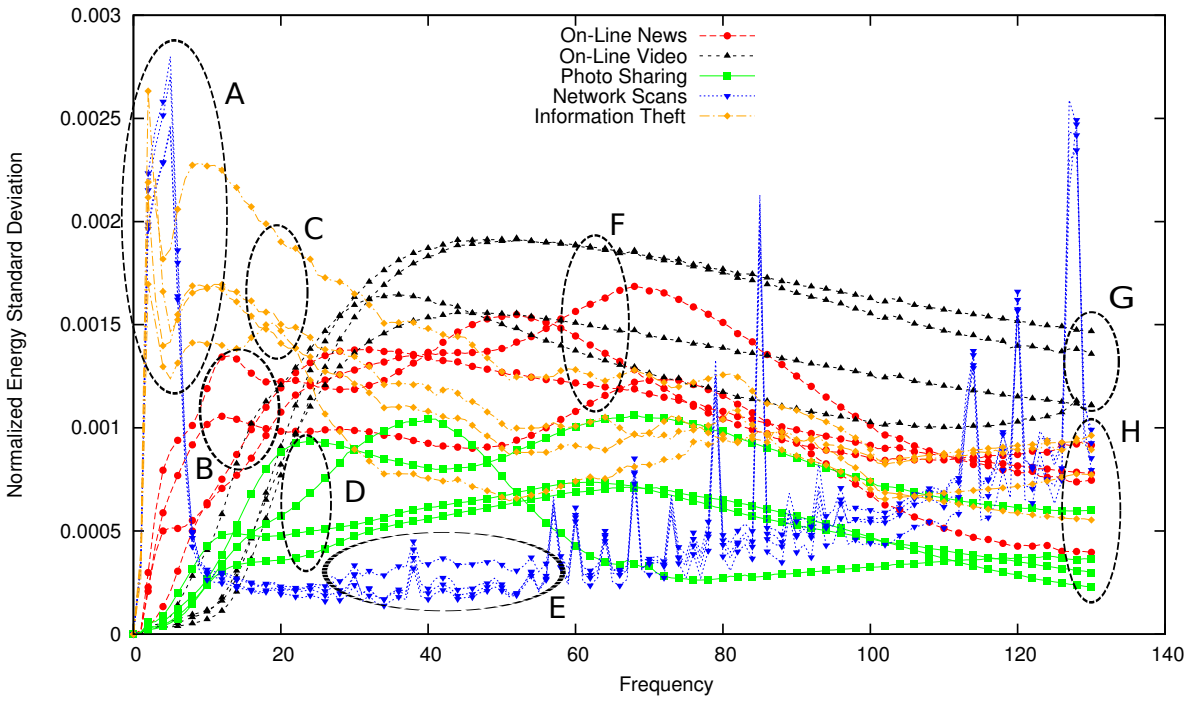

Figure 7.9: Internet Applications and Attacks and corresponding frequency mapping regions.

low number and variation of the created traffic flows. Snapshots can be identified by analyzing other frequency spectrum components in which differentiating regions, such as region C, emerge due to the high energy associated to the creation of file transfer sessions, which are automatic mechanisms associated to user clicks on web-pages. On the other hand, region B was associated to on-line news applications since it encompasses less significant low-frequency events corresponding to user requests typically generated in such services. Region D was associated to photo-sharing applications and includes low-frequency events occurring periodically with low variation since, when using these services, users typically perform periodic requests for downloading other shared images. Region F can be associated to on-line news and video applications since it was created for analyzing the creation of traffic sessions and the presence of traffic control mechanisms. The mentioned services generate significant mid-range frequency components. Finally, regions G and H differentiate applications presenting high and low high-frequency events, respectively. Region G was associated to on-line video services since it comprises applications generating a high amount of network traffic with low IAT. On the other hand, region H includes applications that do not present significant high-frequency components, indicating that they generate a small amount of network traffic. This set includes on-line news, photo-sharing, network scans and snapshots. This classification methodology allowed us to achieve an accurate classification of the traffic generated by the different applications and an accurate identification of some of the most used Internet attacks,

Table 7.3: Internet Applications with their corresponding frequency mapping regions and classification results.

Internet Applications	Regions	Classification Accuracy
On-Line News	B and F and H	**90.00%**
CI		89.00%-91.00%
On-Line Video	F and G	**88.90%**
CI		88.10%-89.70%
Photo-Sharing	D and H	**85.75%**
CI		84.70%-86.80%
Network Scans	A and E	**99.00%**
CI		98.00%-100.00%
Information Theft	A and C and H	**90.00%**
CI		89.20%-90.80%

as shown in table 5.4.1. Some classification overlaps can occur due to similarities between the different classes. This occurred for the On-Line News and Video classes since these are becoming intrinsically connected as many journal web-pages contain embedded videos. On the other hand, some Photo-Sharing traffic was assigned to the News class due to irregular user interactions. Some *data-streams* generated by the emulated attacks were assigned to legitimate application classes due to the stealthiness and low-impact of these attacks. This can cause the illicit traffic to become similar to application classes such as On-Line News.

7.4 Conclusions

This chapter presented an approach for the identification of different web-applications used by distinct clients connected to a wired or wireless network, together with an identification of some important Internet security attacks. By using traffic monitoring and capturing probes, which do not require authentication, we were able to infer layer 2 traffic metrics and perform a Continuous Wavelet Decomposition in order to infer the corresponding traffic scalograms. By analyzing the frequency components present in these scalograms, it was possible to define regions in the frequency spectrum comprising events characteristic to the different web-applications which allowed the accurate identification of the traffic generated by each of the different studied Internet services. By defining user profile as the set of used Internet applications it is possible to build accurate user profiles for different management tasks. The results achieved show that the proposed approach can accurately identify the different web-applications that were run by the connected clients, together with an accurate identification of some Internet-based attacks.

Chapter 8

Conclusions and Future Work

The emergence of the Internet as *de facto* communications platform has lead to the need of accurate user and traffic profiling. In fact, the different services, applications and the ever increasing number of users involved in the Internet has presented several challenges to researchers, Service Providers (SPs) and ultimately to the users themselves. The increasing competition for markets and clients has made SPs increase the capacities of the offered services, which brings a serious repercussion for network management. In addition, the heterogeneity of the Internet applications and their requirements created a need of an accurate mapping of Internet traffic to its originating application(s). Several purposes can be envisioned for such task but the need to perform an efficient management of the network resources and infrastructures, together with the need to timely identify illicit and suspicious traffic have lead to the need of novel methodologies for traffic classification. The increasing capacity of the network links together with the increasing complexity of the Internet ecosystem, where several privacy and technical limitations prevent the analysis of the contents of exchanged traffic, lead to the need for methodologies that can cope with the mentioned restrictions. This thesis addresses this issue by proposing several traffic classification approaches that are suitable to be deployed in scenarios with stringent restrictions.

The recent and alarming increase on the number and variety of Internet attacks has also increased the need for a timely identification of traffic presenting illicit and/or suspicious patterns. In addition, we have also witnessed the emergence of *botnets* as a platform for performing different illicit activities. Due to their distributed architecture, the detection of *botnets* and of the compromised hosts under the control of *bot*-masters has become a complex task that most of existing detection methodologies are not able to accurately accomplish. This also lead to a need for novel *botnet* detection approaches. The traffic classification approaches proposed in this thesis are also suited to accurately detect traffic presenting illicit patterns.

Several scientific contributions were provided in this thesis, which enabled a deeper understanding of the traffic generated by the several existing Internet applications. The main concept behind all the proposed approaches consists on the analysis of the different traffic

generating events and controlling mechanisms. We explored the fact that different Internet applications require different interactions from the user. Such user requests generate a set of Internet sessions, each one creating a set of Internet packets. These are the low (user events), mid (traffic sessions) and high (Internet packets) events that our approaches evaluate. By extracting several traffic metrics and performing a multi-scale decomposition of the captured traffic, our approaches are able to depict the several frequency components present in the captured *streams*. This allows the construction of *Multi-Scale Signatures* for each one of the studied Internet applications. We started by using unsupervised clustering algorithms for grouping *streams* whose multi-scale decomposition estimators present similar variations in the analyzed decomposition scales. Such approach enabled a very accurate and efficient grouping of traffic generated by the same Internet applications, together with an accurate identification of low-impact and stealth attacks. However, the traffic traces used were very long in order to capture sufficient metrics for the clustering algorithm to be able to perform an accurate traffic mapping.

This disadvantage was addressed by using several probabilistic models that enable an accurate association of unknown traffic to the corresponding applications. Such models include unidimensional Gaussian and generic distributions in which each one of the several decomposition scales is analyzed separately to provide traffic classification. Such models were able to provide very accurate traffic classification as well as the identification of some of the most used web security attacks. In addition, the usage of these approaches allowed a reduction on the length of each analyzed *data-stream*. Subsequently, multidimensional models were used to evaluate the correlation between the estimators of the several decomposition scales in order to provide more accurate probabilistic models. All the proposed methodologies enabled an accurate mapping of the analyzed data-streams and the analysis of the mentioned correlations proved to be an important aspect to be analyzed, since the accuracy of the traffic classification approaches was significantly increased. All these probabilistic approaches can be enhanced by using a methodology that performs traffic classification over multiple time-windows. In addition, the analysis of non-sampled traffic metrics allowed a reduction on the size of the classification windows, enabling a real-time traffic classification.

A user-profiling approach for wired and wireless networks was also proposed, together with a monitoring platform where our approaches can be applied. User profiles are defined as the set of web-based applications run by the user and the use of decomposition and identification models presented in sections 3.3.2 and 7.2 enabled an identification of the most significant frequency components present in the traffic of each connected host. In this manner, such components can then be mapped into the application that generated them and an accurate identification of the used web-applications can be performed, enabling an accurate user profiling. Different traffic metrics can be used for analysis according to the scenario and to the existing restrictions. For instance, in wireless networks layer 2 traffic metrics can be

inferred since the deployed monitoring probes do not authenticate with the AP of the monitored network. Therefore, these cannot be detected by any of the connected hosts and can simultaneously monitor clients from different networks.

As future work, the analysis of more probabilistic distributions for our classification methodologies is envisioned. In addition, the use of non-stationary time series representing the several inferred traffic metrics allows the use of several models that can perform a more accurate traffic decomposition, together with a prediction of future values. More stealth and low-impact attacks can be emulated and the analysis of C&C traffic of *botnets* constitute important future work directions. In addition, the development of a platform for monitoring, sampling and analyzing the traffic with our methodologies is also planned, as well as the use of probabilistic approaches for modeling the components present in the different regions of the frequency spectrum for the profiling methodologies. Other decision mechanisms, as well the use of differentiating regions with arbitrary shapes, are also planned. This will provide a more accurate discrimination of the frequency components present in the captured traffic and constitute an important enhancement to the work presented in chapter 7. The use of more traffic metrics and other probabilistic models also constitute an important work that should be conducted in the future.

Bibliography

[ADPG+10] Dmitri Alperovitch, Toralv Dirro, Rahul Kashyap Paula Greve, David Marcus, Sam Masiello, François Paget, and Craig Schmugar. 2010 Threat Predictions. Technical report, McAfee Labs, 2010.

[AFTV00] P. Abry, P. Flandrin, M.S. Taqqu, and D. Veitch. Wavelets for the analysis, estimation, and synthesis of scaling data. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*, pages 39–88. Wiley, 2000.

[AH10] F.H. Abbasi and R.J. Harris. Intrusion detection in honeynets by compression and hashing. In *Telecommunication Networks and Applications Conference (ATNAC), 2010 Australasian*, pages 96 –101, 31 2010-nov. 3 2010.

[ARZMT06] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 41–52, New York, NY, USA, 2006. ACM.

[AT99] Gediminas Adomavicius and Alexander Tuzhilin. User profiling in personalization applications through rule discovery and validation. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 377–381, New York, NY, USA, 1999. ACM.

[BBL10] J. But, P. Branch, and Tung Le. Rapid identification of bittorrent traffic. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 536 –543, oct. 2010.

[BDF+09] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho. Seven years and one day: Sketching the evolution of internet traffic. In *INFOCOM 2009, IEEE*, pages 711 –719, april 2009.

[BMM+07] Dario Bonfiglio, Marco Mellia, Michela Meo, Dario Rossi, and Paolo Tofanelli. Revealing skype traffic: when randomness plays with you. In *Proceedings of the*

2007 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '07, pages 37–48, New York, NY, USA, 2007. ACM.

[BOB+10] H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang. On the analysis of the zeus botnet crimeware toolkit. In *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on*, pages 31 –38, August 2010.

[Bro11] The bro network security monitor. http://www.bro-ids.org/, August 2011.

[BS06] J. Binkley and S. Singh. An algorithm for anomaly-based botnet detection. In *USENIX SRUTI'06*, 2006.

[BSM10] Daniela Brauckhoff, Kave Salamatian, and Martin May. A signal processing view on packet sampling and anomaly detection. In *INFOCOM'10: Proceedings of the 29th conference on Information communications*, pages 713–721, Piscataway, NJ, USA, 2010. IEEE Press.

[BSS10] P. Burghouwt, M. Spruit, and H. Sips. Detection of botnet collusion by degree distribution of domains. In *Internet Technology and Secured Transactions (ICITST), 2010 International Conference for*, pages 1 –8, nov. 2010.

[BTA+06] Laurent Bernaille, Renata Teixeira, Ismael Akodkenou, Augustin Soule, and Kave Salamatian. Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, 36:23–26, April 2006.

[BTS09] Bittorrent: Delivering the world's content. `http://www.bittorrent.com/`, March 2009.

[BY06] P. Barford and V. Yegneswaran. An inside look at botnets. *Springer Verlag*, 2006.

[Cis11] Cisco IOS Intrusion Prevention System (IPS) - Products and Services. `http://www.cisco.com/en/US/products/ps6634/index.html`, March 2011.

[CJM05] Evan Cooke, Farnam Jahanian, and Danny Mcpherson. The zombie roundup: Understanding, detecting, and disrupting botnets. pages 39–44, June 2005.

[CKS+09] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok. A survey on internet traffic identification. *Communications Surveys Tutorials, IEEE*, 11(3):37 –52, quarter 2009.

[CSF+07] M. Patrick Collins, Timothy J. Shimeall, Sidney Faber, Jeff Janies, Rhiannon Weaver, Markus De Shon, and Joseph Kadane. Using uncleanliness to predict future botnet addresses. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 93–104, New York, NY, USA, 2007. ACM.

[Dar08] Darryl veitch. http://www.cubinlab.ee.unimelb.edu.au/ darryl/, December 2008.

[Dar11] 1999 DARPA. `http://www.ll.mit.edu/IST/ideval/data/1999/1999_data_index.html`, August 2011.

[Dau92] Ingrid Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[DCJ10] Xun Dong, John Clark, and Jeremy Jacob. Defending the weakest link: phishing websites detection by analysing user behaviours. *Telecommunication Systems*, 45:215–226, 2010.

[DD08] D. Dittrich and S. Dietrich. P2P as botnet command and control: A deeper insight. In *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*, pages 41 –48, oct. 2008.

[DHKR09] N. Duffield, P. Haffner, B. Krishnamurthy, and H. Ringberg. Rule-based anomaly detection on IP flows. pages 424 –432, apr. 2009.

[Do08] Chuong B Do. The multivariate gaussian distribution relationship to univariate gaussians the covariance matrix. *ReCALL*, M(10):1–10, 2008.

[DP10] Idilio Drago and Aiko Pras. Scalable service performance monitoring. In B. Stiller and F. De Turck, editors, *Mechanisms for Autonomous Management of Networks and Services*, volume 6155 of *Lecture Notes in Computer Science*, pages 175–178, Berlin, June 2010. Springer Verlag.

[DZL06] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using timezones. In *13th Annual Network and Distributed System Security Symposium NDSS'06*, January 2006.

[EAM06] Jeffrey Erman, Martin Arlitt, and Anirban Mahanti. Traffic classification using clustering algorithms. In *MineNet '06: 2006 SIGCOMM workshop on Mining network data*, pages 281–286, New York, NY, USA, 2006. ACM.

[Ell10] Claire Elliott. Botnets: To what extent are they a threat to information security? *Information Security Tech. Report*, 15:79–103, August 2010.

[EpKSX96] Martin Ester, Hans peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[FC09] Nicolas Falliere and Eric Chien. Zeus: King of the Bots. Technical report, Symantec — Security Response, 2009.

[FGW98] A. Feldmann, A. C. Gilbert, and W. Willinger. Data networks as cascades: investigating the multifractal nature of internet wan traffic. In *SIGCOMM '98: ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 42–55, New York, NY, USA, 1998.

[Fre11] Fred-ezone wifi ISP. `http://www.fred-ezone.ca`, August 2011.

[GA05] Daniela Godoy and Analia Amandi. User profiling in personal information agents: a survey. *Knowl. Eng. Rev.*, 20(4):329–361, December 2005.

[GHYC06] Jun Gao, Guangmin Hu, Xingmiao Yao, and R.K.C. Chang. Anomaly detection of network traffic based on wavelet packet. pages 1–5, 31 2006-Sept. 1 2006.

[Gol11] Steve Gold. Advanced evasion techniques. *Network Security*, 2011(1):16 – 19, 2011.

[GPY+07] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. Bothunter: Detecting malware infection through ids-driven dialog correlation. In *16th USENIX Security Symposium*, 2007.

[GWS11] B. Grobauer, T. Walloschek, and E. Stocker. Understanding cloud computing vulnerabilities. *Security Privacy, IEEE*, 9(2):50 –57, march-april 2011.

[GZL08] G. Gu, J. Zhang, and W. Lee. Botsniffer: Detecting botnet command and control channels in network traffic. In *15th Annual Network and Distributed System Security Symposium*, 2008.

[HCL08] Yan Hu, Dah-Ming Chiu, and J.C.S. Lui. Application identification based on network behavioral profiles. *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*, pages 219–228, June 2008.

[HCL09] Yan Hu, Dah-Ming Chiu, and John C.S. Lui. Profiling and identification of p2p traffic. *Computer Networks*, 53(6):849 – 863, 2009. Traffic Classification and Its Applications to Modern Networks.

[HGPS11] J. Hurley, E. Garcia-Palacios, and S. Sezer. Classifying network protocols: A 'two-way' flow approach. *Communications, IET*, 5(1):79 –89, january 2011.

[HJC08] Nen-Fu Huang, Gin-Yuan Jai, and Han-Chieh Chao. Early identifying application traffic with application characteristics. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 5788 –5792, may 2008.

[HSSW05] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. Acas: Automated construction of application signatures. In *MineNet '05: 2005 ACM SIGCOMM workshop on Mining network data*, pages 197–202, Philadelphia, Pennsylvania, USA, August 2005. ACM Press.

[HTS06] Chin-Tser Huang, Sachin Thareja, and Yong-June Shin. Wavelet-based real time detection of network traffic anomalies. In *Securecomm and Workshops, 2006*, pages 1 –7, September 2006.

[IALS11] Jose Antonio Iglesias, Plamen Angelov, Agapito Ledezma, and Araceli Sanchis. Creating evolving user behavior profiles automatically. *IEEE Transactions on Knowledge and Data Engineering*, 99(PrePrints), 2011.

[IAN11] Port numbers. `http://www.iana.org/assignments/port-numbers`, July 2011.

[Ins11] Ponemon Institute. 2010 Annual Study: U.S. Cost of a Data Breach. Technical report, March 2011.

[JG10] Weirong Jiang and M. Gokhale. Real-time classification of multimedia traffic using fpga. In *Field Programmable Logic and Applications (FPL), 2010 International Conference on*, pages 56 –63, 31 2010-sept. 2 2010.

[jKLLK10] Han joon Kim, Sungjick Lee, Byungjeong Lee, and Sooyong Kang. Building concept network-based user profile for personalized web search. In *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on*, pages 567 –572, aug. 2010.

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31:264–323, September 1999.

[KBB+03] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, and M. Faloutsos. File-sharing in the Internet: A characterization of P2P traffic in the backbone. *University of California, Riverside, USA, Tech. Rep*, 2003.

[KBB+04] T. Karagiannis, A. Broido, N. Brownlee, K.C. Claffy, and M. Faloutsos. Is p2p dying or just hiding? [p2p traffic measurement]. *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, 3:1532–1538 Vol.3, Nov.-3 Dec. 2004.

[KBFc04] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc claffy. Transport layer identification of p2p traffic. In *IMC '04: 4th ACM SIGCOMM conference on Internet measurement*, pages 121–134, New York, NY, USA, 2004. ACM.

[KDL04] Oleg Kolesnikov, David Dagon, and Wenke Lee. Advanced polymorphic worms: Evading ids by blending in with normal traffic. Technical report, 2004.

[KPA02] P. Barfordand J. Kline, D. Plonka, and R. Amos. A signal analysis of network traffic anomalies. In *IMW '02: 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 71–82, New York, NY, USA, 2002. ACM.

[KPF05] Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos. Blinc: multilevel traffic classification in the dark. In *SIGCOMM '05: 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 229–240, New York, NY, USA, 2005. ACM.

[KR90] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990.

[KR06] Seong Soo Kim and A.L.N. Reddy. Image-based anomaly detection technique: Algorithm, implementation and effectiveness. *Selected Areas in Communications, IEEE Journal on*, 24(10):1942 –1954, oct. 2006.

[KRH07] Anestis Karasaridis, Brian Rexroad, and David Hoeflin. Wide-scale botnet detection and characterization. In *HotBots'07: First Workshop on Hot Topics in Understanding Botnets*, Berkeley, CA, USA, 2007. USENIX Association.

[KRV04] Seong Soo Kim, A. L. Narasimha Reddy, and Marina Vannucci. Detecting traffic anomalies through aggregate analysis of packet header data. *Networking 2004*, pages 1047–1059, May 2004.

[Lab10] McAfee® Labs. McAfee Threats Report: Fourth Quarter 2010 . Technical report, McAfee Labs, 2010.

[LCD04] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 219–230, New York, NY, USA, 2004. ACM.

[LFG+00] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines, K.R. Kendall, D. McClung, D. Weber, S.E. Webster, D. Wyschogrod, R.K. Cunningham, and M.A. Zissman. Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings*, volume 2, pages 12 –26 vol.2, 2000.

[LG09] Wei Lu and Ali A. Ghorbani. Network anomaly detection based on wavelet analysis. *EURASIP J. Adv. Signal Process*, 2009(1):1–16, 2009.

[Lil67] H. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 1967.

[LTRG09] Wei Lu, Mahbod Tavallaee, Goaletsa Rammidi, and Ali A. Ghorbani. Botcop: An online botnet traffic classifier. *Communication Networks and Services Research, Annual Conference on*, 0:70–77, 2009.

[LWLS06] C. Livadas, R. Walsh, D. Lapsley, and W. Strayer. Using machine learning techniques to identify botnet traffic. In *2nd IEEE LCN Workshop on Network Security*, 2006.

[Lyo09] Gordon Fyodor Lyon. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure, USA, 2009.

[Mac67] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[Mah36] P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55, April 1936.

[MGT+10] Xiaobo Ma, Xiaohong Guan, Jing Tao, Qinghua Zheng, Yun Guo, Lu Liu, and Shuang Zhao. A novel irc botnet detection method based on packet size sequence. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1 –5, may 2010.

[MHLB] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. Flow clustering using machine learning techniques. *Passive and Active Measurement Workshop (PAM2004)*, April 2004.

[MKK+01] David Moore, Ken Keys, Ryan Koga, Edouard Lagache, and K. C. Claffy. The coralreef software suite as a tool for system and network administrators. In *Proceedings of the 15th USENIX conference on System administration*, pages 133–144, Berkeley, CA, USA, 2001. USENIX Association.

[MKR07] S. Majumdar, D. Kulkarni, and C.V. Ravishankar. Addressing click fraud in content delivery systems. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 240 –248, May 2007.

[MM10] Frank McSherry and Ratul Mahajan. Differentially-private network trace analysis. In *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, SIGCOMM '10, pages 123–134, New York, NY, USA, 2010. ACM.

[Mor96] Pedro A. Morettin. From fourier to wavelet analysis of time series. In *In Proceedings in Computational Statistics*, pages 111–122. Physica-Verlag, 1996.

[MP05] Andrew Moore and Konstantina Papagiannaki. Toward the accurate identification of network applications. In Constantinos Dovrolis, editor, *Passive and Active Network Measurement*, volume 3431 of *Lecture Notes in Computer Science*, pages 41–54. Springer Berlin / Heidelberg, 2005.

[MSS+06] Patrick D. Mcdaniel, Subhabrata Sen, Oliver Spatscheck, Jacobus E. van der Merwe, William Aiello, and Charles R. Kalmanek. Enterprise Security: A Community of Interest Based Approach. In *Proc. of Network and Distributed System Security*, 2006.

[MW06] A. Madhukar and C. Williamson. A longitudinal study of p2p traffic classification. *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*, pages 179–188, September 2006.

[MZ05] Andrew W. Moore and Denis Zuev. Internet traffic classification using bayesian analysis techniques. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '05, pages 50–60, New York, NY, USA, 2005. ACM.

[NA06] T. Nguyen and G. Armitage. Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world ip networks. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 369 –376, November 2006.

[NA08] T.T.T. Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. *Communications Surveys Tutorials, IEEE*, 10(4):56 –76, quarter 2008.

[NI07] Aaron Hackworth Nicholas Ianelli. Botnets as a vehicle for online crime. *The International Journal of Forensic Computer Science*, 2(1), 2007.

[NMa11] Nmap: Free security scanner for network exploration and security audits. `http://nmap.org/`, March 2011.

[NMS11] The open NMS project. `http://www.opennms.org/`, June 2011.

[OSS10] Welcome to the home of ossec. http://www.ossec.net/, JUly 2010.

[Pet10] Ivo Petiz. Caracterização de tráfego e comportamentos numa rede p2p-tv. Master's thesis, Universidade de Aveiro, 2010.

[Ram02] A. Ramanathan. Wades: A tool for distributed denial of service attack detection. Master's thesis, TAMU-ECE-2002-02, 2002.

[RFD06] Anirudh Ramachandran, Nick Feamster, and David Dagon. Revealing botnet membership using dnsbl counter-intelligence. In *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*, page 8, Berkeley, CA, USA, 2006. USENIX Association.

[Roe99] Martin Roesch. Snort - lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX conference on System administration*, LISA '99, pages 229–238, Berkeley, CA, USA, 1999. USENIX Association.

[RSN09a] E. Rocha, P. Salvador, and A. Nogueira. Discriminating internet applications based on multiscale analysis. In *Next Generation Internet Networks, 2009. NGI '09*, pages 1 –7, july 2009.

[RSN09b] Eduardo Rocha, Paulo Salvador, and António Nogueira. Detection of illicit traffic based on multiscale analysis. In *Proceedings of the 17th international conference on Software, Telecommunications and Computer Networks*, SoftCOM'09, pages 286–291, Piscataway, NJ, USA, 2009. IEEE Press.

[RSN10] E. Rocha, P. Salvador, and A. Nogueira. Gaussian fitting of multi-scale traffic properties for discriminating IP applications. In *The Second International Conference on Emerging Network Intelligence, EMERGING 2010*, October 2010.

[RSN11a] E. Rocha, P. Salvador, and A. Nogueira. Classification of hidden users' profiles in wireless communications. In *The 3rd International ICST Conference on Mobile Networks and Management*, September 2011.

[RSN11b] E. Rocha, P. Salvador, and A. Nogueira. Detection of illicit network activities based on multivariate gaussian fitting of multi-scale traffic characteristics. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1 –6, june 2011.

[RSN11c] E. Rocha, P. Salvador, and A. Nogueira. A multi-variate classification approach for the detection of illicit traffic. In *Software, Telecommunications Computer Networks, 2011. SoftCOM 2011. 19th International Conference on*, September 2011.

[RSN11d] E. Rocha, P. Salvador, and A. Nogueira. A real-time traffic classification approach. In *The 6th International Conference for Internet Technology and Secured Transactions (ICITST-2011)*, December 2011.

[RSN11e] Eduardo Rocha, Paulo Salvador, and António Nogueira. Can multiscale traffic analysis be used to differentiate internet applications? *Telecommunication Systems*, 48:19–30, 2011. 10.1007/s11235-010-9331-1.

[RSNR11] Eduardo Rocha, Paulo Salvador, António Nogueira, and Joel Rodrigues. Identification of Anomalies on Encrypted Communications for Forensic Purposes. In *The Third International Congress on Ultra Modern Telecommunications and Control Systems*, July 2011.

[RSSD04] Matthew Roughan, Subhabrata Sen, Oliver Spatscheck, and Nick Duffield. Class-of-service mapping for qos: a statistical signature-based approach to IP traffic classification. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 135–148, New York, NY, USA, 2004. ACM.

[Sec10] Symantec Enterprise Security. State of Spam and Phishing. Technical report, Symantec, June 2010.

[Sec11a] Symantec Enterprise Security. Symantec Internet Security Threat Report: Trends for 2010. Technical report, Symantec, June 2011.

[Sec11b] Symantec Enterprise Security. Symantec Report on Attack Kits and Malicious Websites. Technical report, Symantec, April 2011.

[SG10] Alexander Seewald and Wilfried Gansterer. On the detection and identification of botnets. *Computers & Security*, 29(1):45–58, 2010.

[Sha11] Shadowserver foundation. `http://www.shadowserver.org/wiki/`, August 2011.

[SIC11] Sic. http://sic.sapo.pt/, August 2011.

[SM07] Taeshik Shon and Jongsub Moon. A hybrid machine learning approach to network anomaly detection. *Information Sciences*, 177(18):3799–3821, September 2007.

[Sno11] Snort :: Home page. `http://www.snort.org/`, March 2011.

[SSB03] J. Slavic, I. Simonovski, and M. Boltezar. Damping identification using a continuous wavelet transform: application to real data. *Journal of Sound and Vibration*, 262(2):291 – 307, 2003.

[SSW04] Subhabrata Sen, Oliver Spatscheck, and Dongmei Wang. Accurate, scalable in-network identification of p2p traffic using application signatures. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 512–521, New York, NY, USA, 2004. ACM.

[TC98] Christopher Torrence and Gilbert P. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79:61–78, 1998.

[TCP11] Tcpdump/libpcap public repository. `http://www.tcpdump.org`, August 2011.

[TRKN08] Ionut Trestian, Supranamaya Ranjan, Aleksandar Kuzmanovi, and Antonio Nucci. Unconstrained endpoint profiling (googling the internet). In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, SIGCOMM '08, pages 279–290, New York, NY, USA, 2008. ACM.

[TRKN10] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci. Googling the internet: Profiling internet endpoints via the world wide web. *Networking, IEEE/ACM Transactions on*, 18(2):666 –679, April 2010.

[WEK11] Weka 3 - data mining with open source machine learning software in java. `http://www.cs.waikato.ac.nz/ml/weka/`, August 2011.

[WSZ10] Ping Wang, Sherri Sparks, and Cliff C. Zou. An advanced hybrid peer-to-peer botnet. *IEEE Transactions on Dependable and Secure Computing*, 7:113–127, 2010.

[WZ06] Ningning Wu and Jing Zhang. Factor-analysis based anomaly detection and clustering. *Decision Support Systems*, 42(1):375–389, 2006.

[YA09] Wei Yan and N. Ansari. Why anti-virus products slow down your machine? In *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, pages 1 –6, aug. 2009.

[YBP05] Vinod Yegneswaran, Paul Barford, and Vern Paxson. Using honeynets for internet situational awareness. In *In Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets IV*, 2005.

[YR08] Ting-Fang Yen and Michael K. Reiter. Traffic aggregation for malware detection. In *Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, DIMVA '08, pages 207–227, Berlin, Heidelberg, 2008. Springer-Verlag.

[YY94] Akio Yamamoto and Akio Yamamoto. Wavelet analysis: Theory and applications. *Transform*, 46(December):44–52, 1994.

[ZAN99] Ingrid Zukerman, David W. Albrecht, and Ann E. Nicholson. Predicting users' requests on the www. In *Seventh International Conference on User Modeling*, pages 275–284, 1999.

[ZXY+09] Yao Zhao, Yinglian Xie, Fang Yu, Qifa Ke, Yuan Yu, Yan Chen, and Eliot Gillum. Botgraph: large scale spamming botnet detection. In *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 321–334, Berkeley, CA, USA, 2009. USENIX Association.