Traffic monitoring in home networks : from theory to practice

Zied Aouini

UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE
SCIENCES ET INGENIERIE POUR L'INFORMATION

Laboratoire L3I

THÈSE

Présentée par :

Zied AOUINI

Soutenue le 15 Décembre 2017
Pour l'obtention du grade de
Docteur de l'Université de La Rochelle
Discipline: **INFORMATIQUE ET APPLICATIONS**

Traffic Monitoring in Home Networks: From Theory to Practice

JURY:

Mme. Isabelle CHRISMENT	Professeur, TELECOM Nancy, Rapporteur
M. Dario ROSSI	Professeur, TELECOM ParisTech, Rapporteur
M. Philippe OWEZARSKI	Directeur de Recherche, LAAS-CNRS, Examinateur
M. Luca MUSCARIELLO	Ingénieur de Recherche, Cisco Systems, Examinateur
M. Yacine GHAMRI-DOUDANE	Professeur, Université de La Rochelle, Directeur de thèse
M. Absesselem KORTEBI	Ingénieur de Recherche, Orange Labs, Co-directeur de thèse

In memory of my father Monji

Acknowledgements

I am most grateful to my advisors, Dr. Abdesselem Kortebi and Pr. Yacine Ghamri Doudane. Their guidance and insights over the years have been invaluable to me. I feel especially fortunate for the patience that they have shown with me when I firstly stepped into the field of machine learning algorithms applied to network data. I am indebted to them for teaching me both research and writing skills. Without their endless efforts, knowledge and patience, it would have been extremely challenging to finish all my dissertation research and Ph. D study.

It has been a great honor and pleasure for me to do research under their supervision. I would like to thank Pr. Dario Rossi, Pr. Isabelle Chrisment, Pr. Philippe Owezarski and Dr. Luca Muscariello for serving as my Ph. D committee members and reviewing my dissertation. I also owe thanks to Jean Philippe Javaudin, Nicolas Neyret, Jean Yves Cloarec and Christophe Delahaye, who helped me to address technical and scientific problems.

I must thank my family, who supported me a lot. Without their endless love and encouragement, I would have never completed this dissertation.

Finally, I am most grateful to Erij, for standing with me throughout both difficult and good times, and for her love, patience and sacrifices.

Abstract

Home networks are facing a continuous evolution and are becoming more and more complex. Their complexity has evolved according to two interrelated dimensions. On the one hand, the home network topology (devices and connectivity technologies) tends to produce more complex configurations. On the other hand, the set of services accessed through the home network is growing in a tremendous fashion. Such context has made the home network management more challenging for both Internet Service Provider (ISP) and end-users. In this dissertation, we focus on the traffic dimension of the above described complexity.

Our first contribution consists on proposing an architecture for traffic monitoring in Home Networks. We provide a comparative study of some existing open source tools. Then, we perform a testbed evaluation of the main software components implied in our architecture. Based on the experiments results, we discuss several deployment limits and possibilities.

In our second contribution, we conduct a residential traffic and usages analysis based on real trace involving more than 34,000 customers. First, we present our data collection and processing methodology. Second, we present our findings with respect to the different layers of the TCP/IP protocol stack characteristics. Then, we perform a subjective analysis across 645 of residential customers. The results of both evaluations provide a complete synthesis of residential usage patterns and applications characteristics.

In our third contribution, we propose a novel scheme for real-time residential traffic classification. Our scheme, which is based on a machine learning approach called C5.0, aims to fulfil the lacks identified in the literature. At this aim, our algorithm is evaluated using several traffic inputs. Then, we detail how we implemented a lightweight probe able to capture, track and identify finely applications running in the Home Network. This implementation allowed us to validate our designing principles upon realistic test conditions. The obtained results show clearly the efficiency and feasibility of our solution.

Keywords Home Network, Network Performances, Passive Measurements, Traffic classification, Machine Learning Algorithms, Home Gateway, Traffic Analysis, Flow Monitoring.

Résumé

Les réseaux domestiques sont confrontés à une évolution continue et deviennent de plus en plus complexes. Leur complexité a évolué selon deux dimensions interdépendantes. D'une part, la topologie du réseau domestique devient plus complexe avec la multiplication des équipements et des technologies de connecvitié. D'autre part, l'ensemble des services accessibles via le réseau domestique ne cesse de s'élargir. Un tel contexte a rendu la gestion du réseau domestique plus difficile pour les Fournisseurs d'Accès Internet (FAI) et les utilisateurs finaux. Dans ce manuscrit, nous nous concentrons sur la deuxième dimension de la complexité décrite ci-dessus liée au trafic circulant depuis/vers le réseau domestique.

Notre première contribution consiste à proposer une architecture pour la supervision du trafic dans les réseaux domestiques. Nous fournissons une étude comparative de certains outils open source existants. Ensuite, nous effectuons une évaluation de performances expérimentale d'un sous ensemble des processus impliqués dans notre architecture. Sur la base des résultats obtenus, nous discutons les limites et les possibilités de déploiement de ce type de solution.

Dans notre deuxième contribution, nous présentons notre analyse à large échelle des usages et du trafic résidentiel basée sur une trace de trafic réelle impliquant plus de 34,000 clients. Premièrement, nous présentons notre méthode de collecte et de traitement des données. Deuxièmement, nous présentons nos observations statistiques vis-à-vis des différentes couches de l'architecture Internet. Ensuite, nous effectuons une analyse subjective auprès de 645 clients résidentiels. Enfin, nos résultats fournissent une synthèse complète des usages et des caractéristiques des applications résidentielles.

Dans notre troisième contribution, nous proposons une nouvelle méthode pour la classification en temps réel du trafic résidentiel. Notre méthode, laquelle est basée sur l'utilisation

d'un algorithme d'apprentissage statistique de type C5.0, vise à combler les carences identifiées dans la littérature. Ensuite, nous détaillons notre implémentation d'une sonde légère sur un prototype de passerelle résidentielle capable de capturer, de suivre et d'identifier d'une manière fine les applications actives dans le réseau domestique. Cette implémentation nous permet, en outre, de valider nos principes de conception via un banc d'essai réaliste mis en place à cet effet. Les résultats obtenus indiquent que notre solution est efficace et faisable.

Mots-clés Réseau domestique, Performances réseau, Mesures passives, Classification du trafic, Algorithmes d'apprentissage statistique, Passerelle domestique, Analyse du trafic, Supervision des flux.

List of Publications

International conferences (published)

Full papers

Aouini, Z., Kortebi, A., & Ghamri-Doudane, Y. (2015, August). Traffic monitoring in home networks: Enhancing diagnosis and performance tracking. In International Wireless Communications & Mobile Computing Conference, IWCMC 2015. IEEE.

Aouini, Z., Kortebi, A., & Ghamri-Doudane, Y. (2016, November). Towards understanding residential internet traffic: From packets to services. In International Conference on Network of the Future, NOF 2016. IEEE.

Kortebi, A., Aouini, Z., Juren, M., & Pazdera, J. (2016, October). Home Networks Traffic Monitoring Case Study: Anomaly Detection. In Global Information Infrastructure and Networking Symposium, GIIS 2016. IEEE.

Short papers

Kortebi, A., Aouini, Z., Delahaye, C., Javaudin, J. P., & Ghamri-Doudane, Y. (2017, May). A platform for home network traffic monitoring. In IFIP/IEEE Symposium on Integrated Network and Service Management, IM 2017. IEEE/IFIP.

International conferences (under submission)

Aouini, Z., Kortebi, A., & Ghamri-Doudane, Y. (Submitted). Early Classification of Residential Networks Traffic using C5.0 Machine Learning Algorithm. In International Conference on Innovation in Clouds, Internet and Networks (ICIN 2018). IEEE.

National Conferences (published)

Aouini, Z., Kortebi, A., & Ghamri-Doudane, Y. (2016, July). La Supervision Du Trafic Dans Le Réseau Domestique : Amélioration Du Diagnostic Et Du Suivi De Performances. In

13ème conférence Francophone sur les nouvelles technologies de la répartition, NOTERE 2016. IEEE.

Patents

Aouini, Z., Kortebi, A. (2015, December). Procédé de détermination d'une application génératrice d'un flux IP. INPI registration number: FR1563301

Journal articles (under preparation)

Aouini, Z., Kortebi, A., & Ghamri-Doudane, Y. Towards Understanding Residential Networks' Usages: From Packets to Customers

Aouini, Z., Kortebi, A., & Ghamri-Doudane, Y. Early Classification of Residential Networks Traffic using C5.0 Machine Learning Algorithm: From Theory to Practice.

Contents

List of Figures

List of Tables

List of Abbreviations

ADSL	Asymmetric Digital Subscriber Line
BRAS	Broadband Remote Access Server
BSSID	Basic Service Set Identifier
CDN	Content Delivery Networks
CISDA	Computational Intelligence for Security and Defense Applications
CPU	Central Processing Unit
CQR	Communications Quality and Reliability
DDoS	Distributed Denial of Service
DNS	Domain Name System
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
FN	False Negative
FP	False Positive
FPGA	Field-Programmable Gate Arrays
FTTH	Fiber To the Home
GUI	Graphical User Interface
HNA	Home Network Assistant
HNID	Home Network Infrastructure Devices
HNMC	Home Network Monitoring Center
IA	Internet Accounting
IANA	Internet Assigned Numbers Authority
IE	Information Elements
IETF	Internet Engineering Task Force
IM	Integrated Network Management
IoT	Internet of Things
IP	Internet Protocol
IPFIX	IP Flow Information eXport
IPRED	Intellectual Property Rights Enforcement Directive
IPTV	Television over IP
ISP	Internet Service Provider

IWCMC	International Wireless Communications and Mobile Computing Conference
LAN	Local Area Network
LFE	Learning Features Extraction
MAC	Media Access Control
MLA	Machine Learning Algorithms
NAS	Network Attached Storage
NIC	Network Interface Card
NTP	Network Time Protocol
OLT	Optical Line Termination
OS	Operating System
OSI	Open Systems Interconnection
OTT	Over-the-top
PLC	Power Line Communication
QoE	Quality of Experience
QoS	Quality of Service
QUIC	Quick UDP Internet Connections
RTFM	Real Time Traffic Flow Measurement
RTT	Round-Trip-Time
SCTP	Stream Control Transmission Protocol
TCP	Transmission Control Protocol
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol
VoD	Video on Demand
VOIP	Voice over IP
WAN	Wide Area Network
WG	Working Group

Traffic Monitoring in Home Networks: From Theory to Practice

Chapter 1 Introduction

1.1 Overview of Home Networks

Home networks are facing a continuous evolution and are becoming more and more complex. Their complexity has evolved according to two interrelated dimensions. On the one hand, the home network topology (devices and connectivity technologies) tends to produce more complex configurations [1]. On the other hand, the set of services accessed through the home network is growing in a tremendous fashion. Such context has made the home network management more challenging for both Internet Service Provider (ISP) and end-users.

Let us introduce the context where the home network is placed today, with respect to the two-point perspective (topology and services) introduced above.

From a topological perspective, the home network tends to be more extensive and composite as there are more devices connected using heterogeneous technologies. In fact, the home network is the interconnection of the Home Gateway (Access Gateway) with the user's end-devices set. The continuous reduction of Central Processing Unit (CPU) costs according to the Moore law [2] has driven the proliferation of the users' connected devices. As example, an average of 6.8 screens per French household is recently reported in [3]. Indeed, the user's end-devices set has evolved from a single computer to a large set of various terminals such as Laptops, Tablets, Smartphones, Smart TVs, Gaming consoles, Network Attached Storage (NAS), etc. In addition, the observed growth of Internet of Things (IoT) devices [4] enlarge this set with new devices such as smart body scales, smart door locks, connected thermostat, etc. Additionally, Home Network Infrastructure Devices (HNID) such as Ethernet switches, Power Line Communication (PLC) plugs, Wi-Fi access points, Wi-Fi extenders and so on are used to expand the coverage to the whole house. A subset of these HNIDs could even induce hybrid links (i.e. Wi-Fi/PLC). Consequently, the home network topology is evolving

from a classic star based topology (around the Home Gateway) to a more complex tree or mesh topology [1].

From services perspective, the evolution is twofold. On the one hand, the proliferation of new devices has generated a new spectrum of ISP's offered services. Indeed, classical managed services (i.e. Voice over IP (VOIP), Television over IP (IPTV) or Video on Demand (VoD)) are continuously enriched with new usages and services such as Energy management, Healthcare or Home Monitoring and Control which emerged as promising business opportunities in the residential market. On the other hand, Internet services and applications have witnessed a boom during the last two decades. In fact, web activities were limited to visiting some text and images contents hosted by a savvy set of servers during the early days of Internet. Nowadays, web activities face an exponential growth resulting on dozen to hundreds of modern web applications and services such as media streaming portals (e.g. Netflix, YouTube, DailyMotion, Spotify, etc.), online gaming platforms (e.g. Steam, Xbox Live, etc.), communications services (e.g. Google Hangout, Facebook Messenger, Viber, Skype, etc.) or social networks (e.g. Facebook, Twitter, Instagram, LinkedIn, etc.) [5].

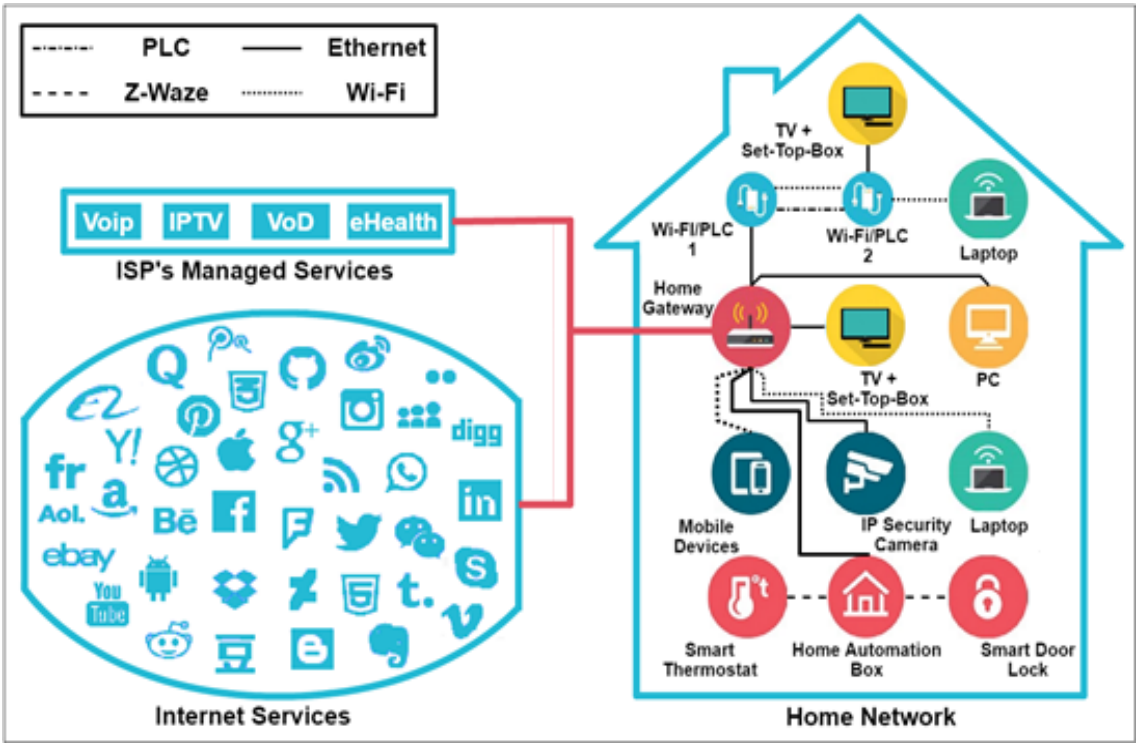

Figure 1.1 Home Network Services and Hybrid Connectivity Technologies

Figure 1.1 depicts a modern home network context where several devices are connected through hybrid technologies to the Home Gateway. The Home Gateway interconnects the

home network with the Wide Area Network (WAN) access. Thus, both managed services and web services can be accessed.

While understanding the above introduced complexity on a single schema is tricky, the question that arises logically is: How both end-users and ISPs deal with such configuration in real case scenarios?

From an end-user perspective, the home network management appears as a challenging task. For instance, a user can start a video conference while another one is downloading a big file. Both services require bandwidth and the quality perceived by both users may be affected. In this scenario, users can hardly determine the cause of their performance degradation issue. In fact, authors in [6] report the difficulties that the household members face when troubleshooting, maintaining, and setting up their home network. In [7], authors highlight the fact that end-users were even unable to articulate properly what the problem is. Based on a subjective study conducted in both US and UK, the experiment shows that the home network management is challenging even for advanced users. Thus, authors conclude that a great potential exists for developing applications that help end-users while managing their home network.

From an ISP standing point, when the customers are bothered by some issues with their home network, they tend to call the ISP hotline. The ISP's diagnosis process in such context is both costly and frustrating. In fact, discussions with several large access ISPs are reported in [8] and reveal that hotline calls are costly, ranging from 9 to 25$ per call. Furthermore, 75% of hotline calls from customers are usually caused by problems that have nothing to deal with the ISP. Thus, improving the diagnosis process with tools providing more visibility of the user's home network context is mandatory to improve customer's satisfaction level and to reduce the hotline cost by decreasing both frequency and duration of hotline calls. Finally, it will help ISPs to stand out in a highly competitive market where regulatory agency become more and more interested in comparing the quality of service offered by ISPs with what they actually deliver [9].

1.2 Problem Statement

As discussed above, managing efficiently the home network portion, yields to several benefits for both users and ISPs. To perform this task, both above dimensions of home network complexity (topology and services) must be tackled. While tools providing a full topology,

discovery are addressed in [10] and start to be deployed by ISPs, we focus in this thesis on providing a complete visibility of home network traffic. In particular, we aim to provide both end-users and ISPs with a real-time monitoring system of active applications running in the home network. Consequently, tracking each application performances (e.g. transmitted bytes, throughput, etc.) will facilitate the diagnosis process for both end-users and ISP help desks. Additionally, it will allow the ISP to deploy new Quality of Service (QoS) prioritization and management mechanisms. For instance, a user can prioritize Skype flows over other application flows or block a given application in a specific time range for parental control purposes (e.g. blocking Social Networks applications on the kids' devices after 9pm). Furthermore, we highlighted in [11] how traffic monitoring is mandatory in several home networks for anomalies detection scenarios (i.e. flooder device detection, blocking Distributed Denial of Service (DDoS) attacks, etc.). Finally, a fine characterization of home networks traffic helps ISPs to adapt their infrastructure according to customers' usages and hence, leverage new billing opportunities.

As stated, a crucial step when facing the challenge of home networks management is a real-time monitoring of running applications and their corresponding performances. To perform such task, two main processes must be achieved:

a) Performance monitoring: capture, monitor and track performance metrics (bytes/packets transferred, Transmission Control Protocol (TCP) Flags, duration, etc.) of the active flows in the Home Network.
b) Application Identification: Identify the application (e.g. Facebook, Skype, Google, etc.) behind each monitored flow.

1.2.1 Performance monitoring

Deploying a monitoring probe able to perform both described processes is already challenging in backbone networks where servers and substantial infrastructure are dedicated at this aim [12]. In a Home Network context, this task become more challenging as several questions must be addressed. The first question that arises logically is: where the probe should be placed?

Unless deploying dedicated hardware in each household which is an unrealistic approach, there is mainly two possible approaches: the end-host based approach [13, 14, 15, 16] and the Home-Gateway based one [17, 18]. Each approach has its pros and cons.

The end-host based approach consists of placing a monitoring unit on each end-host device. Monitoring the Home Network applications performances at the closest point (i.e. the device running the application) appears as the most accurate solution. However, such architectural choice is assuming having control of all user's end devices. In real-life scenario, only few devices in the home network are controlled by the ISP (typically the Home Gateway, TV decoder and some HNIDs) which limits the viability of such an approach. Another common weak spot of the end-host based approach that bias the diagnosis results is the un-observability of activities of other devices which do not run the corresponding probe. Due to the above-mentioned reasons, some research efforts focused on an alternative solution.

The Home Gateway based approach overcomes the lack of network visibility described above. As it constitutes a central point of the home network, monitoring major part of traffic activity is achievable by placing a monitoring unit on the Home Gateway. Moreover, the Home Gateway is fully controlled by the ISP and thus, such approach is more viable from an ISP standing point. Nevertheless, such placement strategy is challenged by several hardware constraints that had not been addressed in the past. In fact, the ISP's Home Gateways are typically limited in terms of CPU, memory and storage capacity. Moreover, traffic monitoring is mainly based on the flows' packets observation which is not trivial to achieve on such devices. In fact, Home Gateways are designed to support packets routing at high speeds (typically packets switching and routing are performed by dedicated Field-Programmable Gate Arrays (FPGAs) (also known as hardware accelerators)). Consequently, classical packet observation methods (packet observation at the Operating System (OS) kernel space) are not applicable. In this dissertation, we focus on designing a Home Gateway based real-time monitoring architecture. Thus, the above introduced limitations must be addressed carefully.

To tackle the constraints related to the Home Gateway based application monitoring approach, we need a deep understanding of how a monitoring system is designed. There are two main categories of monitoring approaches: the active and the passive one. On the one hand, the active monitoring approach consists on actively monitoring a dedicated metric by injecting traffic in the network. On the other hand, the passive monitoring approach consists on computing performance metrics based on the network traffic observation.

As discussed earlier, we aim to perform more than a single metric supervision and thus, we focus in this thesis on passive approaches. Also, among all possible approaches, we argue in this thesis that the flow export architecture fulfils a large set of home network monitoring requirements. It mainly consists of a probe which observes packets, aggregates them into

flows and exports continuously records of the performed observations to an external component for collection and analysis. In this thesis, we propose to deploy flow export architectural components in a Home Network context. Therefore, several questions arise naturally regarding the feasibility of such approach:

a) Does the overhead (CPU, memory and network load) induced by such approach is sustainable in a Home Network context?
b) How do we perform packets observation if packets are not observable (hardware accelerator concerns)?

To sum up, the implemented probe must achieve packet observation, flow aggregation and export with a limited resource impact. Furthermore, it must deal with the requirement of our second process: the application identification.

1.2.2 Application identification

Application identification is a corner stone in our approach as we aim to provide an application oriented monitoring system. To identify an application, we need to classify the traffic belonging to this application's flows. While integrated to our architecture, the addressed challenge is fourfold. First, we must identify the application behind each flow at an early stage (real-time constraint for QoS or parental control usage). Second, our classification output granularity must be fine enough to distinguish specific services (e.g. Facebook, Skype, Bittorrent, etc.). Third, our approach must be resilient to traffic evolution (e.g. encryption, tunneling, patterns changes, etc.). Finally, the overall classification process must be performed in a lightweight manner to cope with the Home Gateway characteristics discussed constraints. Thus, our scope is related to the wide research field of traffic classification.
The oldest and most classical used approach for traffic classification is the port-based one. It consists on mapping the observed communication ports in the transport protocol header to the well-known labels assigned by the Internet Assigned Numbers Authority (IANA). Despite being fast and lightweight, a major drawback of this approach is its unreliability due to several reasons (e.g. port abuse, random port usage with Peer to Peer (P2P) applications for instance, tunneling, etc.). Moreover, the resulting classification granularity (e.g. Hypertext Transfer Protocol (HTTP), Domain Name System (DNS), etc.) is completely outdated with respect to the context of modern web activities. As discussed above, such context implies a need of fine grained classification. The first alternatives tackling the inadequacy of port-based

approaches relied on payload inspection for packets pattern matching [19, 20]. These methods, usually called Deep Packet Inspection (DPI) techniques, examine the content of each packet of a flow looking for a set of characteristic signatures. Despite being accurate, DPI technique induces a consequent computational overhead. Moreover, its accuracy is challenged by encrypted traffic trend (i.e. HTTPS, HTTP 2.0, Quick UDP Internet Connections (QUIC) by Google, etc.). Furthermore, virtual tunneling technologies (including Tor) are evolving rapidly and are more adopted by residential users due to several factors (e.g. privacy concerns, European Union Intellectual Property Rights Enforcement Directive (IPRED) and the HADOPI law [21] in France). Finally, a heavy signatures engineering task to cope with traffic evolution is needed.

To overcome these issues, Machine Learning Algorithms (MLA) emerge as an alternative. They rely on identifying statistical patterns of applications based on a set of flows characteristics used to train a given algorithm. A high overall accuracy is reported using MLA in a large subset of the literature [22, 23, 24, 25, 26, 27, 28, 29, 30, 31]. However, several gaps limit the deployment of these approaches [32]. In this thesis, we rely on a MLA approach to identify Home Network applications. Consequently, the commonly identified issues from the literature must be addressed. Thus, the resulting solution must be lightweight, real-time, fine-grained and resilient to traffic evolution.

1.3 Contributions

Taking into consideration the constraints imposed by the Home Network context, this dissertation proposes an architecture for real-time traffic monitoring. Our architecture is mainly based on a probe that is able to capture, identify and track applications running in the Home Network. To achieve this goal, several challenges are addressed. In a nuttshell, this thesis makes contributions to two interrelated research fields which are flow monitoring and traffic classification.

Our contributions consist, firstly, on designing a novel architectural approach to perform home network flow monitoring. To understand the deployment possibilities and limits, we conducted a comparative study of existing flow monitoring open source tools. Then, we evaluated a promising one (nProbe [33]) on an experimental testbed focusing on resource consumption criteria. The obtained experimental results were positive in terms of resource consumption as well as bandwidth utilization for typical Digital Subscriber Line (DSL) access

speed scenario. However, our work highlighted several possible improvements and challenges. We leverage the need of a reliable traffic identification method. While we evaluated a widely used DPI library using several configurations, we showed that the overhead induced by such approach is quite low. And thus, we concluded that the main lack of DPI method is the need of a heavy signature engineering in addition to encrypted traffic concerns. Therefore, we turned our interest towards overcoming such issue using an MLA approach.

Secondly, we focus on characterizing residential traffic and usages. In fact, residential traffic characterization is a key aspect for ISPs to tune up their networks according to their customers' requirements. Despite the significant gap between business and residential customers, a large body of literature measurements is performed at higher observation points and only few studies have examined residential traffic characteristics. Furthermore, traffic analysis granularity is often too coarse to tackle observed growth of applications and services. In this contribution, we present a fine-grained analysis of a real residential traffic dataset collected in France and provided by a major ISP involving more than 40,000 customers. Moreover, we conducted a subjective behavioral analysis of 645 residential customers. The benefits of this contribution are twofold. While our findings provide useful insights of residential usage patterns and applications characteristics, the collected data is also used as a starting point to our MLA approach design.

Thirdly, we propose a fine-grained early classification method for residential traffic. Our approach main core is based on the C5.0 machine learning algorithm trained to identify modern Internet services. At this aim, we relied on our previously collected dataset and developed the suitable tools to process it. Our solution achieves an average accuracy of 98.8% while finely classifying applications flows (i.e. Facebook, Google Services, Skype, BitTorrent, Web-Browsing and Secure-Web-Browsing) using statistical features of the very first packets of each flow. Performances are evaluated using advanced metrics based on a disjoint testing dataset involving more than 34,000 residential customers. Consequently, we think that our results are more convincing than the previously reported ones based on a synthetic single user dataset. Moreover, we provide the community with an extension which, integrated with open source components, allows a reliable data processing chain. Finally, we ensure the viability of our approach by proposing a retraining architecture to address MLA deployment issue.

Finally, we present a home network traffic monitoring platform. We implemented a probe on a home gateway prototype (having the same chipset and hardware characteristics as a commercially deployed one) that is able to capture the traffic and export performance metrics at real-time. The probe performs also real-time traffic classification using our previously developed machine learning approach. Consequently, the classifier allows application flows to be identified based on the statistical features of the very first packets only. Our design principles overcome the hardware accelerators issue that is inherent to this kind of devices. Finally, our implemented probe is evaluated using several scenarios. Our experimental results are promising and prove a limited impact in terms of resource consumption (CPU, memory and network load). Thus, we conclude that despite several possible improvements, our proposed approach is fulfilling a large part of residential traffic classification stated issues and challenges.

1.4 Thesis Structure

The rest of this dissertation is structured as follows.

In the second chapter, we provide an overview of prior works related to the research areas we highlight in this thesis. First, we expose strengths and weaknesses of existing architectural approaches. Then, we deepen our overview and focus on flow monitoring architecture. Finally, we present an analysis of existing application identification approaches. We focus on MLA approaches and depict the lacks related to this field.

In Chapter 3, we introduce our proposed architecture for traffic monitoring in Home Networks. We provide a comparative study of existing open source tools. Then, we perform a benchmark of the nProbe tool and evaluate the computational overhead of each performed process. Based on experiments results, we discuss several deployment limits and possibilities.

In Chapter 4, we present our conducted residential traffic and usages analysis. First, we present our data collection and processing methodology. Second, we present our finding with respect to TCP/IP protocol stack characteristics. Then, we conduct a subjective analysis across 645 of residential customers. Finally, our findings provide a complete synthesis of residential usage patterns and applications characteristics.

In Chapter 5, we propose a novel scheme for real-time residential traffic classification. Our scheme, based on the C5.0 decision tree algorithm, aims to fulfil the lacks identified in the literature. At this aim, our scheme is evaluated using several traffic inputs. Then, we detail how we implemented it as part of a lightweight probe able to capture, track and identify finely applications running in the Home Network. Hence, our design principles are validated using a real testbed.

Finally, in Chapter 6, we conclude this dissertation by summarizing our main contributions and discussing some perspectives.

Chapter 2 Traffic Monitoring and Classification in Home Networks: Approaches, Concepts and Limitations

2.1 Introduction

In this thesis, we focus on providing both users and ISP with a real-time monitoring of running applications in the Home Network and their corresponding performances. Such automated system will facilitate the management and diagnosis processes and thus, benefit for both ISP and customers.

Real-time monitoring of active applications in the home network involves monitoring active flows and mapping each of them to its corresponding application in a reliable manner. Thus, our scope falls in the intersection of two wide but complementary research fields which are Traffic Monitoring and Traffic Classification. In this chapter, we review the literature in each of both research field. Our aim is to place prior works and existing concepts with respect to the Home Network context. We focus on four functional requirements:

- Full visibility: To diagnose and troubleshoot a Home Network, the monitoring approach must provide a full visibility of the network activities.
- Real-time flow monitoring: The monitoring approach must be real-time oriented and must be able to monitor active flows present in the Home Network.
- Early and Reliable application identification: The timeliness and the reliability of how an approach identifies the application behind each flow is a main concern. First, applications must be identified at early time (real-time identification) to allow online actions

such as parental control, anomalies detection, QoS management, etc. Second, Applications must be identified in a fine manner and thus allow to distinguish the real application behind the flow such as Facebook or Bittorrent. Finally, the monitoring approach must be resilient to traffic evolution (encryption, new protocols, etc.)

- Computational and hardware limitations: In a Home Network context, the monitoring approach must be lightweight as it will be naturally deployed on a Home Network connected device (Home Gateway, HNIDs or a user end device). Our definition of 'lightweight' includes both computational load (i.e. CPU and memory) and network load (bandwidth overhead induced by running the solution). Moreover, the monitoring approach must deal with hardware constraints (hardware accelerators) of the Home Gateway when its placement strategy involves such a device.

Studying existing concepts and approaches according to the above requirements allows us to explain in detail where our work comes at play.

2.2 Active vs. Passive Monitoring Approaches

Network monitoring approaches have been well studied and developed throughout the years. A starting classification could be the used mode to perform the measurements.

2.2.1 Active monitoring approaches

Active monitoring approach consists on measuring a dedicated metric by injecting traffic in the network. Several tools were previously proposed such as:

- Ping and traceroute.
- King [34]: estimates delay by measuring the delay between the closest DNS servers.
- Pathload [35]: measures available bandwidth.
- T-rat [36]: evaluates the rates at which flows transmit data.

Despite being accurate, the required measurement overhead is a concern to our objective which is supposed to operate in a continuous manner. Furthermore, this approach deals only with metrics collection. It does not apply to active flow monitoring neither to application identification. Consequently, we turn our interest on passive monitoring approaches for traffic monitoring.

2.2.2 Passive monitoring approaches

Passive monitoring approaches consist on observing traffic at an observation point to extract traffic information and performance metrics. One passive approach is based on packet capture. This method consists of a full packet capture, storage and analysis. The major drawback of this approach is an expensive hardware and the substantial infrastructure need for storage and analysis. In our home network context, this approach is naturally non-viable, as we aim to use a Home Network connected device (which have generally limited resources (even if it's a commercial connected PC)).

Another passive network monitoring approach that fits better our context is flow export, in which a probe aggregates packets into flows and export observation records to a collector for storage and analysis. Flow Export technologies like Cisco NetFlow [37] or IETF standardized IP Flow Information eXport (IPFIX) [38] collects Internet Protocol (IP) traffic information, such as source and destination IP addresses, ports, timestamps for the flow start and finish time, number of bytes and packets observed in the flow and so on. This approach is powerful to collect IP information statistic and provide flexibility that makes it easy to extend. As it offers distributed configuration alternative (probes/collector on different devices), flow export appears as a suitable solution to our purposes.

2.3 Home Network Monitoring Architectural Approaches

As previously depicted in Figure 1.1, the home network architecture could be summarized as the interconnection of the home gateway with the users' end devices. Additional infrastructure devices (e.g. Ethernet switches, Wi-Fi access points, etc.) can also be used. Consequently, achieving a complete visibility of the residential traffic is possible through two architectural approaches. The first one is to place a monitoring unit per End-host device, whereas the second one is to deploy one monitoring unit on an ISP's controlled device, typically the home gateway.

2.3.1 End host based approaches

To improve troubleshooting and to ensure a better understanding of home network usage, some research efforts were oriented on the end-host approaches. A monitoring unit has been

designed in several works to run on the end user devices. As the aim is to collect different network information, the usage of this collection process differs from one project to another. In the following we describe few of these propositions.

2.3.1.1 HostView

HostView [13] is a data collection utility that runs on individual end-hosts (running Linux and MAC OS). It exports the following information from the hosting device:

a) Network data:
- Packet headers (anonymized IP source)
- Extract content-type from HTTP responses (Image or text, video, audio)
- Full DNS packets (with anonymized local IP addresses)
- Log periodically applications (process names) associated with open network sockets based on *gt* toolkit [39].
- Log the active network interface type: wired or wireless
- Records the hash of the Media Access Control (MAC) address of the home gateway for an Ethernet connection or the hash of the Basic Service Set Identifier (BSSID) of the access point for a wireless connection
- A pop-up questionnaire asks the user to describe the networking environment (i.e. home, work, airport, coffee shop)

b) Machine performance data: the sysperf module ensures measuring system performance metrics such as CPU load.

c) User feedback data: the user feedback module incorporates two different mechanisms:

- An "I am annoyed!" button that is always displayed at the edge of the screen; users are supposed to click on it when they are not happy with their network performance.
- A system-triggered feedback form, which prompts the user three times per day to respond to a questionnaire about their network experience in the 5 minutes preceding the questionnaire. The system-triggered questionnaire is configurable. The user can turn it off.

All the above data is logged on the user machine and periodically uploaded to a remote server (at LIP6-UPMC). The server then transfers the data sets to a back-end repository disconnected from the Internet. Finally, the HostView tool provides the user with some network usage statistics (per device overall bandwidth consumption, per application bandwidth usage, etc.).

2.3.1.2 HomeMaestro

HomeMaestro [14] is a distributed system running on multiple end devices for the monitoring and the instrumentation of home networks. It performs extensive measurements at the host level to infer application network requirements, and identifies network related problems through time-series analysis.
Monitoring processes implemented to extract network statistics on End-hosts (Windows Vista OS) include:

- Monitoring read/write operations at the network socket level
- Monitoring internal TCP state of all connections
- Collecting extensive measurements (TCP's estimation of the RTT, total number of bytes/packets IN/OUT, congestion events)
- Collecting application-specific information such as process name and used libraries.

By sharing and correlating information across hosts in the home network, the system automatically detects and resolves contention over network resources between applications (e.g. limiting flow's rate to resolve contention on constrained resources such as the upstream of the Internet access link) based on predefined policies. Finally, HomeMaestro implements a distributed virtual queue to enforce those policies by prioritizing applications without additional assistance from network equipment such as routers or access points.

2.3.1.3 *Netalyzer*

The Netalyzr [16] tool analyses various properties of the Internet connection. Those properties include:

- Blocking of important services: direct TCP access to remote servers (HTTP, IMAP, SNMP, NetBIOS, etc.) status (Blocked or allowed) and possible reasons explanation.
- HTTP caching behavior and proxy correctness,
- DNS server's resilience to abuse,

- NAT detection
- Latency & bandwidth measurements.

The results are presented in a detailed report form. To perform these tests, the Netalyzer tool runs a Java applet on the user device (Multiplatform tool).

2.3.1.4 HomeNet Profiler

The HomeNet Profiler [15] is a tool that runs on any computer (Windows, MacOS, Linux) connected inside a home network. It collects a wide range of measurements about home networks including the set of devices, the set of services (with UPnP and Zeroconf), the characteristics of the Wi-Fi environment (ESSID, BSSID, channel number and RSSI), running and installed application's information and end-host configuration information (OS name, OS version, etc.)
The HomeNet Profiler runs one-shot measurements upon user demand and integrates the Netalyzer module to enrich statistics. Finally, it lists the running and installed applications without mapping them to the corresponding flows.

Table 2.1 Positioning of studied end-host approaches

Reference	Full visibility	Real-time flow monitoring	Early and Reliable application identification	Computational/hardware limitations
HostView		✓	✓	Not addressed
HomeMaestro		✓	✓	Not addressed
Netalyzer				Not addressed
HomeNet Profiler			✓	Not addressed

2.3.1.5 Discussions and positioning

As summarized in Table 2.1, end-host studied approaches do not fulfil our stated functional requirements. In fact, these tools have been developed in a measurement and data collection logic. Consequently, several lacks arise with respect to our defined requirements. Our first observation is regarding the full visibility criteria. Indeed, the proposed approaches is assuming having control of each of customer's devices. As deployment on devices such as laptops or smart phones would be tricky (multiple OS, client agreement, etc.) but still possible, the increasing number of connected devices and objects (smart TV, smart sensors, etc.) makes this deployment approach non-viable from an ISP standing point. Moreover, these tools suffer from the un-observability of activities of other devices inside the home network (e.g. guest devices), which do not run the corresponding tools, which can bias the diagnose results.

Secondly, studied approaches do not address resource consumption limitations despite considering deployment on constrained resource devices (Smartphone, tablet, etc.). Our explanation is that the studied tools are mainly designed for research data collection purposes and thus, they are not optimized for low resource consumption but rather for high accuracy investing more resources.

Note that only two tools among the studied ones perform both real-time flow monitoring and a reliable application identification. We focus on application identification used method. In particular, we consider identifying applications on end-host with a direct mapping between the process and the active network socket as an asset for end-host architectural placement. In fact, such method is identified in the literature [40] as the most accurate and reliable one.

For these reasons, we are more interested in ISP controlled devices based approaches. In particular, the home gateway appears as the most appropriate monitoring point as it is the border between the Local Area Network (LAN) and the WAN constituting a central point of the home network from where a full visibility could be achieved (except for intra LAN traffic which can be addresses through intermediate nodes).

2.3.2 Home Gateway based approaches

As major part of the traffic (going in/out the home network) passes through the home gateway, this approach appears as the most promising to fulfil our full visibility criteria. A first idea to perform this approach would be to directly apply the existing end-host based solutions on home gateways. However, this is not possible due to various reasons. Firstly, the home gateway is typically constrained in terms of resources and hardware design. Secondly, end-host measurements (e.g., applications process name or network stack details) are not achievable on home gateways where traffic packets are the only available source of data.
Due to the above-mentioned restrictions, some research efforts on how home network monitoring could be performed from a home gateway perspective have been conducted.
Calvert et al. [41] consider requirements for a general-purpose logging facility for home networks. They propose to capture packets events using tcpdump and to log wireless (L2)-related events using *ewenet*. Their study concludes that such strategy will miss up to 10% of traffic due to home gateway buffer size restrictions. Moreover, storing the data on such device is considered as a major bottleneck.

To overcome data storage limitation, the Bismarck [18] project proposed a home gateway firmware (based on OpenWRT) which is extended by a flow monitoring function. The Bismarck firmware includes the Bismarck-passive function which passively collects traces including flow and packet records (timestamps, size, ports, IP addresses, transport protocol and IP to domain name mappings from DNS traffic) and exports them to a remote server. Several works are based on the Bismarck firmware to perform further analysis. For instance, authors in [8] reuse the Bismarck firmware to design the "Where is The Fault" home gateway-based solution. Authors aim to identify if the home network performance degradation is due to the wireless link or the access link. In [17], authors studied the feasibility of application performance tracking on home gateways, which involves both identification of active applications and monitoring their performance. They implement a modified version of the Bismarck firmware to perform some additional metrics measurements. However, real-time traffic identification is not formally addressed. Finally, The EU projects Nanodatacenters [42] and Figaro [43] design the home gateway for Next-generation Internet services and include traffic monitoring as a key component. They consider flow export technology as a suitable candidate for their architecture. However, authors neither consider the hardware limitations nor the resource consumption constraints introduced above.

2.3.2.1 Discussion and positioning

Studying the different existing solutions, the Bismarck project and flow export based solutions appear the closest to fulfil our requirements. Firstly, both solutions are home gateway based and thus, achieve partly our full visibility criteria. In fact, some intra LAN traffic such as traffic between hosts not connected directly to the home gateway (e.g. through intermediate HNIDs like an Ethernet switches) may not be observed. While such lack was not addressed in prior works, we propose in chapter 3 a home network monitoring architecture that considers such scenarios.

Secondly, the Bismarck firmware runs on WRT compatible routers and is designed for resource constrained devices. However, the overhead induced by enabling the Bismarck passive flow monitoring function (without application identification) is evaluated in [17] and is pointed as unsustainable in terms of CPU and memory consumption when forwarding up to 96Mbps traffic. Moreover, authors conclude that real-time traffic identification is not feasible on home gateways and propose to deploy this function on an external server. In our context, our monitoring approach must be lightweight and must be sustainable at 1Gbps speed. Additionally, traffic identification must be real-time and must run on the home gateway. In fact,

we consider that placing such process on an external component eliminates the real-time criteria. Indeed, recognizing an application after the flow termination is useless for blocking/prioritization scenarios.

For the above-mentioned reasons, we focus on flow export methods. Our interest is turned to the IETF IP Flow Information eXport (IPFIX) standard [38]. Designed as a flexible standard, it offers large configuration possibilities. However, several challenges must be tackled to fulfil our defined requirements. In fact, previous studies based on such technology do not address the hardware constraint of home gateways. In our work, we evaluate the resource overhead induced by running such technology on a real home gateway. Additionally, we aim to overcome hardware acceleration limitation which raises the challenge to higher levels. Moreover, prior works do not address the traffic identification step. Worse, authors in [17, 44] claim that the Home Gateway can observe and export flows only in a best-case scenario where traffic identification is placed on a remote dedicated server. In this thesis, our contributions are oriented to achieve traffic observation, export and a reliable identification of applications in a lightweight manner. To achieve these goals, we need a deep understanding of both flow export architecture and traffic classification concepts. In the rest of this chapter, we provide some insights studying the literature of both fields.

2.4 Detailed Overview of the IPFIX Architecture

2.4.1 Historical background of the IPFIX standard

While the first publication of IETF IPFIX was in 2013, its standardization process is the result of flow monitoring evolution during preceeding two decades. Figure 2.1 details this evolution back to the 90s. The different steps were:

1991 Internet Accounting (IA) Working Group (WG) of the Internet Engineering Task Force (IETF) (Started on 1990) describes aggregation of packets into flows based on packet header information (Concluded in 1993).

1995 The authors of [45] presented a methodology for profiling traffic flows based on packet aggregation.

1996 IETF Real Time Traffic Flow Measurement (RTFM) WG started with the objectives of producing an improved traffic flow model and developing an architecture for improved flow measurements. Cisco worked on its flow export technology named NetFlow and patented it the same year.

1999 IETF RTFM WG published a generic framework for flow measurements (RTFM Traffic Measurement System).

2002 NetFlow first version, NetFlow v5, became available to public.

2004 Cisco releases NetFlow v9 with an improvement in terms of flexibility. IETF starts a standardization process of flow export protocol (IP Flow Information eXport WG).
2006 Cisco provided more flexibility publishing Flexible NetFlow.
2008 First IPFIX specification.
2011 Cisco presented NetFlow Lite based on Flexible NetFlow.
2013 IPFIX internet standard published.

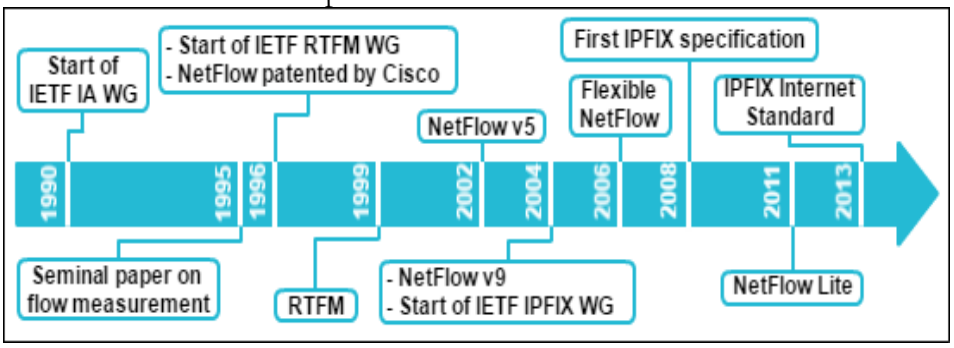

Figure 2.1 Historical evolution of IPFIX standard

Flow export approach faced a lack of vendor's interest until Cisco published NetFlow v5 and released NetFlow v9 two years after. As NetFlow v9 data model was freely available, IETF selected NetFlow v9 as the starting candidate to its IPFIX standardization process. While NetFlow technologies are still widely deployed and integrated into ISP's high-end packets forwarding devices (e.g. routers, firewalls, etc.), we observe that vendors and ISP's are progressively integrating IPFIX in their solutions. On one hand, IPFIX standardization's process strongly involved researchers (e.g. Fokus [46], ETH Zürich [47] and WAND [48]), industry actors (e.g. Cisco) and operators (e.g. NTT) resulting on various IPFIX open source implementations [49, 33]. On the other hand, NetFlow v9 was criticized for its lack of flexibility. IPFIX extends monitoring process to new devices, systems and usages and thus, several networking actors (e.g. Cisco, Nortel, Dell, Juniper, FlowMon Networks, nTop, etc.) start to adopt IPFIX in their solutions.

2.4.2 IPFIX architecture

As we study the feasibility of constrained resource devices flow monitoring based approach, we focus on different IPFIX architecture components. The aim is to identify possible deployment bottlenecks and limits. Furthermore, it is a key step to compare existing IPFIX toolsets. Figure 2.2 illustrates the IPFIX architecture where several stages are depicted. The main two components of the IPFIX architecture are the probe (exporter) and the collector.

The IPFIX probe achieves packets observation, aggregation into flows and export of observations records. The collector collects the data exported from the probe and analyze it. IPFIX consider the probe as a device. Such definition includes dedicated devices (e.g. servers) which are connected through a network tap or a port mirroring configuration. In our context, the probe must be a home network connected device, since deploying a dedicated probe device in each household is not viable. Such configuration leverages the weight of our computational limitation challenge as the probe device is not fully dedicated for monitoring task and thus, our approach impact must be as transparent as possible. For the above reasons and for sake of brevity, we focus more on probe processes rather than collector one.

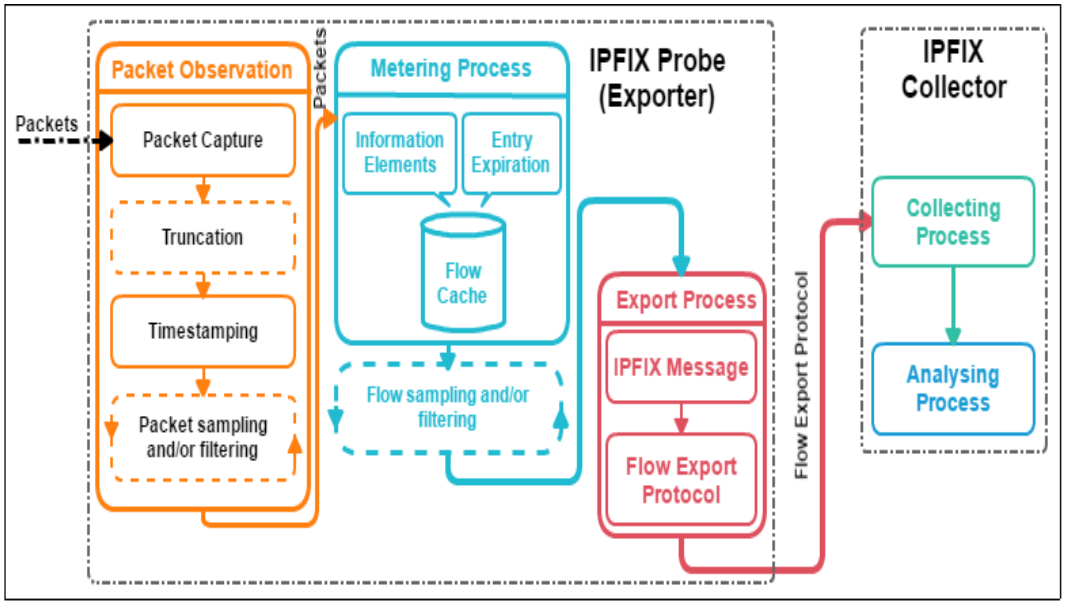

Figure 2.2 IPFIX overall architecture

2.4.2.1 Packet observation

Packet observation is a key stage in flow monitoring as it is the starting point. Consequently, we detail in the following each step involved at this phase:

Packet capture: This step is performed on the Network Interface Card (NIC) level. After passing various checks such as checksum error, packets stored in on-card reception buffers are moved to the hosting device memory. Several libraries are available to capture network traffic such as libpacap; libtrace [50] for UNIX based operating systems and Winpcap for Windows. These libraries are running on the top of the operating system stack which may reduce performances passing through several layers. To overcome such limitation in a high-speed network context, software and hardware optimization technique are proposed:

- Network stack bypass techniques which consist on using a zero-copy path while transmitting packets data from NIC buffers to userspace. While several libraries exist such as PF_RING/ZC [51], netmap [52], Intel DPDK [53], a question that naturally arises and that we address in chapter 3 is: what are the benefit of using such technologies in a home network context?
- Cost effective hardware capture optimization is provided by some vendors of commodity NICs. It consists of hardware acceleration of packet capture process. In this case, packet capture (and processing in some cases) is/are directly performed by dedicated FPGAs. Consequently, the CPU load is not impacted by the capture process. Such design choice is deployed on modern home gateways for networking functions (forwarding, routing, etc.) purposes. However, monitoring traffic while such hardware components are present is an open research question. In fact, authors in [54] highlighted packets offloading techniques (which include hardware accelerators) as open issues in the traffic monitoring field. A first solution that could be considered is to implement a monitoring probe as an additional function of those FPGAs. Such strategy is not viable for several reasons. First, such technology is usually vendor proprietary and thus, ISPs do not have access to such option. Second, such FPGA are not designed for this task and are very constrained in terms of resources capacity. Finally, a monitoring probe is a highly evolving software component. Application monitoring requires a frequent update cycles to deal with traffic evolution. Consequently, such software characteristic is not applicable on a hardware low level implementation context. In this thesis, our aim is to design and implement a probe which overcomes the introduced limitations.

Timestamping: As packets may come from several observation points, reordering process is based on packet's timestamp. While hardware timestamping provides a high accuracy up to 100 nanoseconds in case of the IEEE 1588 protocol, it's not supported by most of commodity NIC. Software timestamping based on Network Time Protocol (NTP) is widely used to outcome this lack providing an accuracy up to 100 microseconds [55].

Truncation (optional): Defining a snapshot length, the process selects precise bytes from the packet. It is performed in some cases to reduce the amount of data captured by the probe and therefore CPU and bus bandwidth load.

Packet sampling (optional): is generally performed to reduce load on subsequent stages. It can be systematic (periodic sampling scheme) or random. The latter is recommended as periodic scheme may introduce unwanted correlation in the observed network data.

Packet filtering (optional): performs filtering of packets to separate packets having specific properties from those not having them [56]. A packet is selected if some specific fields are equal or in the range of given values. Another technique is a hash based filtering, applying a hash function on a portion of the packet, the result is compared to a value or a range of values.

2.4.2.2 Metering process

It includes packets aggregation into flows and flow records exporting process. A flow layout is defined as a set of Information Elements. First, the metering process performs the aggregation of packets into flows based on its layout (set of Information Elements). Second, a flow entry is cached until the flow is considered as terminated (entry expiration). Finally, optional steps such as flow sampling and filtering may be performed.

Table 2.2 Common IPFIX information elements

ID	Name	Description
152	flowStartMilliseconds	Timestamp of the flow's first packet
153	flowEndMilliseconds	Timestamp of the flow's last packet
8	sourceIPv4Address	IPv4 source address in the packet header
12	destinationIPv4Address	IPv4 destination address in the packet header
7	sourceTransportPort	Source port in the transport header
11	destinationTransportPort	Destination port in the transport header
4	protocolIdentifier	IP protocol number in the packet header
2	packetDeltaCount	Number of packets for the flow
1	octetDeltaCount	Number of octets for the flow

Information Elements: IPFIX flow record's fields are named Information Elements (IEs). It can be divided into two categories; IANA IE's registry that ensures cross-vendor interoperability and enterprise-specific IEs that allow defining new IEs for particular uses. IE is defined by numeric ID (added to an enterprise ID in enterprise-specific IE case), name, length (fixed or variable), type and status. IPFIX suggests that IE can be defined on any layer level going from layer 2 to layer 7 (application awareness using application identification techniques). In addition to that, IPFIX includes some SNMP MIB information IEs to prevent redundant definition of IEs on the IANA registry. Finally, few advanced mechanisms have been defined to handle IEs:

- Variable length encoding which can be used for variable length IE to avoid waste of bytes due to fixed length IEs.

- Structured data [57] is useful for encapsulating a list of the same IE in a single field (MPLS labels, for example). In IPFIX, probe instruments collectors by means of a template which is a description of used IEs. Table 2.2 describes the minimum IEs set needed to describe a flow.

Flow Caches: Flow caches consist of tables in which the metering process stores information regarding active flows in the network. A flow key (a set of IEs, typically IP source and destination addresses, source and destination ports, and protocol) determines whether a packet is matching an existing flow entry in the cache or not. In the first case, flow's counters are updated. In the latter one, a new entry is created. Non-key fields are utilized to collect flow characteristics (e.g. number of packets, number of bytes, etc.). If IP addresses are part of flows key, and that traffic between two pairs generates flows on both directions; IPFIX provides bidirectional flows records solution adding counters for both directions and special IEs such as "biflowDirection" that indicate the flow initiator pair and reverse. The cache's size depends on exporter device memory capacity and should be configured based on criteria such as key/non-key fields, maximum number of flows expected and expiration policy.

Entry expiration: Cache entries are maintained in the cache table until they are considered as terminated. Termination of a flow is triggered by an expiration event. According to IPFIX, the metering process should consider an entry as expired based on:

- Natural expiration: observed TCP packet belonging to a flow with FIN/RST flag.
- Emergency expiration: flush a certain number of entries to free some space when the cache become full.
- Active timeout: a flow entry expires after being considered active during a certain period (range from 120 seconds to 30 minutes). Counters are reset while start/ end timestamp are updated.
- Idle timeout: a flow entry expires if no packets belonging to it are observed during a specific period (range from 15 second to 5 minutes).
- Resource constraints: special heuristics such as dynamic timeouts configuration at runtime.
- Cache flush: flush of all the entries due to unexpected situations.

It is possible to configure our metering process based on expiration policy to reduce the amount of records exported.

Flow record Sampling and Filtering: Flow record sampling and filtering processes are quite like packet sampling and filtering process explained above. The major differences are the processed unit; while packet sampling and filtering process packets, flow sampling and filtering process flow records coming from the metering process.

2.4.2.3 Export process

The export process involves the following components:

IPFIX message: Figure 2.3 shows a simplified version of the IPFIX message format [38]. Field size (in bytes) is indicated for fixed size fields. The first 16 bytes constructs the header of the message. One or multiple sets described by an ID and a variable length come after the header. It can be typed from one of the following:

- Template sets: contains one or more templates that describe the layout of data records.
- Data sets: used to handle data records (flow records).
- Options template: are used to send metadata to collectors (example: a probe informs the collector which flow keys are used by the metering process).

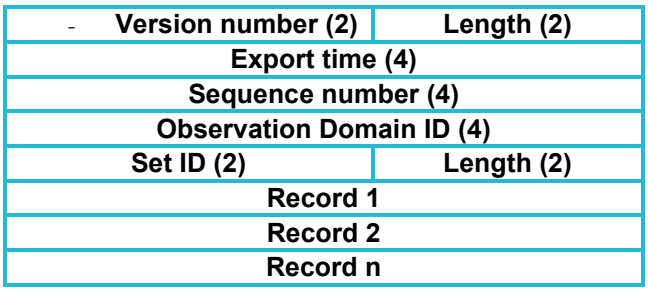

Figure 2.3 Simplified IPFIX message

The exporting process decides if a set is composed from one or multiple records (usually a limited number of records to avoid IP fragmentation). While IPFIX message size does not exceed maximum MTU (1500 bytes) of a link [38], an exception may arise when using variable length IEs. Figure 2.4 shows a detailed example composed by: template, corresponding data record and flow records.

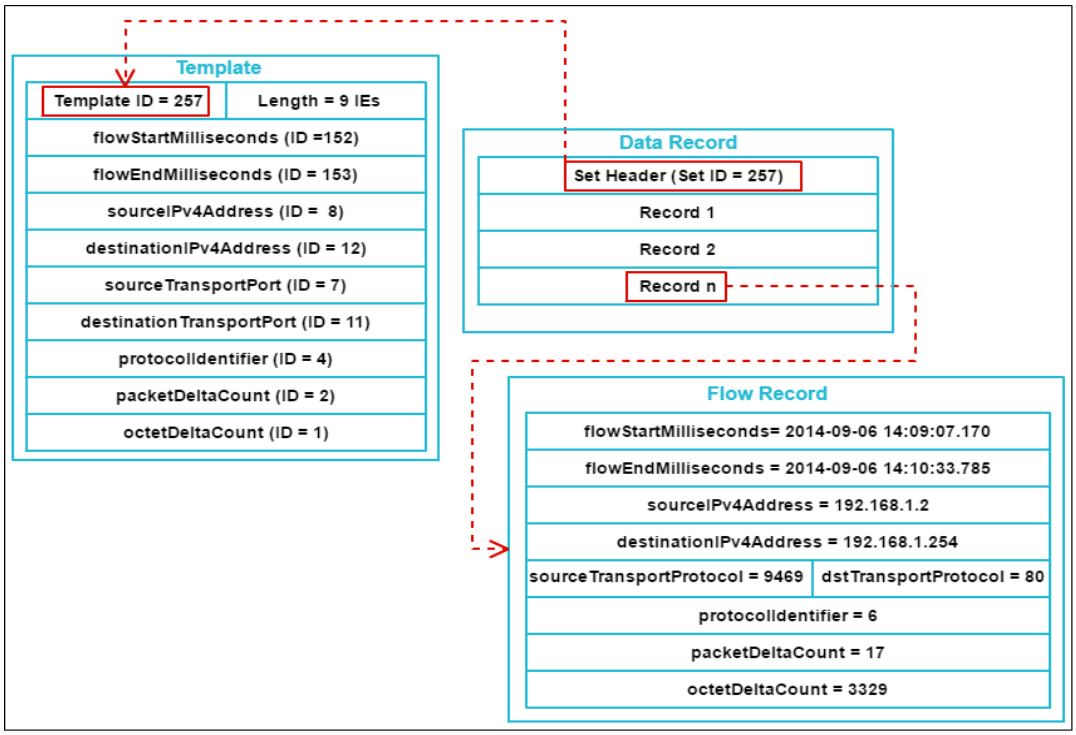

Figure 2.4 Correlation between IPFIX data types

Transport protocol: A crucial step in the flow monitoring export is to select an adequate transport protocol which must be supported by both nodes running the probe and the collector instances. TCP, User Datagram Protocol (UDP) and Stream Control Transmission Protocol (SCTP) are the supported protocols while the latter is the mandatory protocol to implement. In fact, SCTP provides a considerable advantage assuring a compromise between TCP and UDP features and providing more flexibility [58]. After a certain timeout, IPFIX export process over SCTP is able to cancel retransmission of unreceived datagrams which avoid overloading the export buffer with pending retransmissions ensuring a graceful degradation. One major drawback of SCTP is that it is still the least deployed one due to compatibility issue with existing systems.

As mentioned, IPFIX over TCP is supported as well. While TCP stack is continuously improved and widely deployed which makes deployment of IPFIX support over TCP easy, it doesn't provide a graceful degradation in overload situations. In fact, the collector (TCP receiver) window mechanism limits the sending rate of the probe (TCP sender) in such situations and results on the probe buffer saturation.

Finally, IPFIX over UDP is still the most widely deployed and implemented. Despite being unreliable providing a "best effort" service, UDP is the leader on IPFIX current implementation.

2.4.2.4 Data collection process

Data collection process deals with reception of messages sent by the probe at the collector side, storing them in an adequate manner, and pre-process data (compression, anonymization [59], aggregation [60], filtering and summary generation) before transmitting them to the analysis module.

Storage Formats: The collector performance depends strongly on how the storage process is performed. We have two types of storage formats:

- Volatile: very fast as it is performed in memory. Generally used in real time processing approach as a caching step where only results had to be stored (generating time-series to be stored on persistent format).
- Persistent: performed for long time data storing. It can be considered as a second level. Persistent storage is significantly slower than the volatile one. Speed and performance depend on storage type: flat files, row-oriented databases and Column-oriented databases.

Data anonymization: Flow data are less privacy sensitive than raw packet traces as they exclude packet payload but still exploitable by hackers to identify individuals and track their activity. To this concern, data anonymization process [61] could be performed at collector side to protect customer's data. A survey of data anonymization techniques is provided in [62].

2.4.2.5 Data Analysis process

The results of all previously explained stages are provisioning a final stage which consists of data analysis. Data analysis could be classified according to three application areas.

Flow analysis and reporting: In our approach, flow monitoring probe would be deployed in strategical locations (typically home gateways or HNIDs) which would allow to gather a comprehensive set of home network connections summaries. Flow analysis and reporting current basic functionalities are:

- Browsing and filtering flow data.
- Statistics overview: detailed statistics such as top talkers in the network, statistics per host, Autonomous Systems or services.

- Reporting and alerting: a common reporting functionality is bandwidth reporting (which user/host exchanged how much traffic). Alerting process is generally configured to trigger an alert when specified thresholds are exceeded (e.g. number of connections per host, communication with unwanted protocols or applications, etc.).

Threat Detection: Flow data is used for threat detection operations. We can distinguish between two types of uses:

- Use of flow data: flow export is useful for detection of attacks and malwares such as DDoS attacks, worn spreading, network scan and botnet communications. In fact, these attacks typically affect metrics that could be extracted from flow records (number of active flows during a time interval, volume of traffic in terms of packets and bytes, suspicious port numbers or destination hosts). In general, suspicious destination hosts are based on a reputation lists or a blacklist used as reference to compare host destination in the flow records.
- Use of the definition of the flow: this approach is based on the common definition of a flow to detect certain types of attacks.

Performance monitoring: Performance monitoring observes the status of running services in the network. Typical observed metrics are Round-Trip-Time (RTT), delay, jitter, response time, packets loss and bandwidth usage. Flow based performance monitoring applications post-process flow data and presents a set of metrics per target service.

2.4.3 Discussions and Positionning

As detailed above, IPFIX architecture is composed of a set of several processes on both probe and collector sides. In this thesis, our goal is to deploy IPFIX based monitoring system in a home network context. Consequently, sevral questions must be addressed regarding the deployment of such architecture. In fact, we need to identify accurately where possible bottlenecks may appear, and at which level. Is it the packet observation, the metring or the export process?

On a second step, we need to evaluate the real benefits of the mentioned optimization techniques. For example, software packet capture optimization had proven its efficiency in a high-speed network context but has not been evaluated in the home network context.

Moreover, IPFIX probe exporting process induces an additional network load. Such additional load must be studied as our goal is to monitor tens of millions of Home networks.

In this thesis, we address the above questions in Chapter 3 while evaluating the nProbe tool.

Finally, the classification method used by the exporter to identify the application flows is not fixed by IPFIX. The used method must deal with several challenges inherent to traffic classification field.

2.5 Traffic Classification Approaches

As IPFIX architecture provides flexible definition of IEs, we focus in this section on the following step: application identification (traffic classification). In fact, one of our functional requirements is a reliable application recognition (mapping an active flow to its corresponding application). We note that IPFIX provides IEs definition to export/collect application information without recommending how to process the recognition process. Application classification techniques could be regrouped according to two input data types:

- End-host data input: it is the most reliable and accurate approach. It consists on directly mapping the open network socket of a flow to the process name [39, 63] . Such technique is the most lightweight and is not impacted by the traffic characteristic change neither by the encryption trend. However, it is achievable when the probe is directly placed on the end-host. Consequently, it inherits the above architectural option stated lacks.
- Traffic data input: it consists on extracting from packets belonging to a given flow the knowledge required to identify the originating application. Such approach is more suitable for Home Gateway based placement. However, the reliability and the computational induced overhead is correlated to the used technique. In the rest of this section we focus on detailing the state of the art of the existing techniques to this data type.

2.5.1 Port based approach

The most common method (and oldest) for traffic classification is the port-based approach which consists of mapping the used communication ports observed in the TCP/UDP header to the well-known TCP/UDP port numbers assigned by the IANA [63, 64]. As port numbers are easily accessible, this approach provides the advantage of being fast and having low computational cost. Nevertheless, such approach reliability is heavily affected by modern applications characteristics:

- Port abuse: Applications like P2P can use non-standard ports for communication (e.g. BitTorrent can run on TCP: 80 if all ports are blocked).

- Random port usage: Applications can use random ports for communication (e.g. BitTorrent can run on any TCP or UDP port configured by the user)
- Tunneling: Applications can tunnel traffic inside other applications to prevent detection.

Moreover, the obtained granularity (HTTP, DNS, SMTP, etc.) is too coarse with respect to our needs and thus, such approach is considered as outdated.

2.5.2 Deep Packet Inspection

Deep Packet Inspection (DPI) was proposed to address port based approach drawbacks. It is based on the inspection of the content of packets beyond layer 4 headers, searching for distinctive hints of application protocols in the packets payload. The payload is searched for signatures (keywords, known patterns, regular expressions) which are specific to an application protocol. This approach is implemented in various commercial solutions (e.g. NBAR and NBAR2 by Cisco, ixEngine by Qosmos, etc.) as well as in open source projects such as OpenDPI [65], nDPI [19], L7-filter [66], libprotoident [20].

The question that arises naturally is: Are these techniques reliable for classifying modern traffic?

In [67], the authors studied the accuracy of DPI solutions based on real collected traffic traces. While DPI approach is known as providing high accuracy, this study showed that only two libraries which are libprotoident and nDPI could be considered as reliable and provide the best accuracy among open source libraries. In [68], the results of the previous work are confirmed using a larger collection of traffic data. Moreover, only nDPI output is fulfilling our fine-grained definition of the classification output and is able to distinguish services such as Google, Facebook, etc.

The second question that arises is: What is the cost in terms of resource consumption while enabling such technology?

DPI techniques had been initially criticized for being resource consuming [69, 70]. However, recent implementations such as libprotoident and nDPI are less resource consuming. Moreover, devices running these kinds of tools are becoming more powerful. In chapter 3, we aim to answer this question while evaluating resource consumption of nDPI using real traffic on a testbed developed at this aim.

DPI approach suffers from another major drawback which is the signatures engineering. In fact, signatures database must be continuously updated and deal with traffic evolution which leads to high engineering costs.

Another limitation is that, DPI accuracy is challenged by encrypted traffic trend. Furthermore, virtual tunneling technologies (including Tor) are evolving rapidly and are more and more adopted by residential users due to several factors (e.g. privacy concerns, European Union Intellectual Property Rights Enforcement Directive (IPRED) and the HADOPI law [21] in France). Finally, deep packet inspection is considered as illegal in some countries due to the privacy concerns of using such intrusive technique.

2.5.3 Machine Learning Based Approaches

To overcome the above mentionned issues, Machine Learning Algorithms (MLA) emerge as an alternative. It consists in detecting, in an offline phase (usually called Training Phase), characteristic patterns of the applications based on a set of flows' statistical observations. More specifically, machine learning algorithms use a training dataset which is a collection of flow features to extract and then generate the knowledge in a specific output structure (e.g. clusters, rules, decision trees) depending on the used algorithm. During the online phase, the obtained structure is used to classify unlabeled flow features while assuming that the extracted knowledge is sufficiently representative to recognize the statistical behavior of a given application. The large body of literature in this field could be categorized into two main areas: the supervised [71, 24, 25, 27, 28, 29, 72, 73, 74] and unsupervised [71, 22, 23, 26, 27, 30, 31] learning approaches. While the first category requires a labelled dataset predefining the output classes, the latter discovers hidden structures from unlabeled data. A high overall accuracy (over 90%) is reported using MLA approach in a large subset of the literature [22, 23, 24, 25, 26, 27, 28, 29, 30, 31].

As introduced above, a fine classification granularity is a key requirement in modern networks context and only a subset of the proposed approaches fulfil this requirement. Indeed, MLA seems to represent the best option to cope with our needs; therefore, in the rest of this dissertation, we will focus on this approach. Note that, MLA reliability depends heavily on the methodology used to collect and to process the training dataset as reported in [32]. Especially, the performance of the supervised approach relies directly on the quality of the input label. Furthermore, some proposed approaches are based on the post-mortem flow statistical features (i.e. transferred bytes/packets, cumulative TCP flags, packets size distribution, etc.).

Despite being highly accurate, such timeliness is not relevant in a real-time management/control context (i.e. QoS management, Layer 7 policer, etc.). Finally, traffic characteristics evolve rapidly (software updates, end-to-end encryption trend, new OTT delivery protocols such as QUIC, etc.) which decreases the performance achieved by an MLA approach at a given time, if no updates are performed. Hence, the retraining process of MLA is a crucial step to ensure temporal robustness. Previous works tend to provide a "one shot" evaluation and such issue has not been formally addressed.

2.6 State of the Art of MLA approaches: Limitations and Positioning

Traffic classification is a well-studied research field. In fact, several MLA based approaches had been proposed during the last decade and are surveyed in [75, 76]. In [77], Khalife et al. provided a descriptive taxonomy for the properties of traffic classification methods. Based on this defined taxonomy, we depict in Table 2.3 several MLA based approaches used as landmarks during the design of our solution. We focus particularly on seven characteristics to compare the studied methods.

2.6.1 Input Features and Early Classification

A large set of traffic features is used as input for an MLA approach as listed in [78] and could be categorized into two branches as per their observation level. The first one [22, 26, 72] is based on features computed at packet level (i.e. size, inter-arrival delay, etc.), while the latter [23, 25, 28, 29, 30, 73, 74] uses flows post-mortem statistics (i.e. transferred bytes/packets, cumulative TCP flags, packets size distribution, etc.). Finally, some propositions are based on a combination of both levels [71, 24, 27, 31] to perform higher accuracy.

The early identification aspect refers to the timeliness of a proposed approach. While some approaches [22, 26, 27, 72] are able to provide early classification based on first packets characteristics, a major part of contributions [71, 23, 25, 24, 28, 30, 31, 73, 74] is performed at post-mortem (after flow termination) stage. Note that the timeliness of an approach is directly related to the used input features (packet: early, flow: post-mortem).

2.6.2 Dataset

We focus on the characteristics of used dataset as it is the corner stone of an MLA approach. First, we distinguish between synthetic datasets generated by some tools [24, 29, 31, 73] and the ones captured from live networks [71, 22, 23, 24, 25, 26, 27, 28, 30, 74]. In fact, performances reported in a small lab network are not generalizable to a real network context. Worse, it is even not generalizable from a large campus network as pointed out in [32]. Nevertheless, most of real traffic datasets are collected in campus networks. In fact, ISP real datasets are rarely published, and some published ones are truncated which limits the ground-truth generation process.

One key factor while gathering the data, is obtaining a reliable ground-truth. While it is easily achievable on synthetic dataset (i.e. socket name captured on end-host) [73], such process is much more complicated on real datasets. Early works [71] used port-based approach which is well-known for its unreliability. Researchers tried to overcome this gap by using DPI tools as ground truth generators. Such approaches [22, 23, 24, 25, 26, 27] had been considered as reliable until authors in [40] raised the unreliability of most commonly used libraries. In fact, comparative studies of several DPI engines showed that only two engines [19, 20] over six could be considered as providing a reliable accuracy. Consequently, we consider ground truth by nDPI [19] and libprotoident [20] as trustworthy. We also include in this category methods that are based on end-host labelling.

2.6.3 Encryption Awareness

Despite the wide spread of encrypted application protocols, some approaches [23, 24, 25, 26, 28, 30] do not deal with such traffic classes. Furthermore, Secure Shell protocol (SSH) is the most commonly used example for encryption aware approaches validation. From our point of view, we strongly believe that modern usages require a deeper granularity (services behind SSL/TLS) when dealing with encryption challenges.

2.6.4 Output Granularity

The output granularity definition has evolved according to the continuously growing set of modern web applications. Nowadays, a fine classification is defined as an approach able to identify the application/service behind a given protocol. Only [74] fulfils this definition. In fact, other surveyed works provide a coarse definition of used output classes (FTP, P2P, HTTP, etc.). On one hand, this observation could be explained by the limitation of the used

ground-truth generation tool. On the other hand, defining large classes limits the error rate of an approach. Consequently, our classification approach output must be fine-grained (Facebook, Skype, Google-Services, etc.) to fulfil this requirement.

2.6.5 Machine Learning Methods

While both ML methods (supervised and non-supervised) provide good performance, the main difference is the need of labelled data in the supervised ones [71, 24, 25, 27, 28, 29, 72, 73, 74]. Such labelled data implies the predefinition of output classes of the supervised approach. Conversely, non-supervised approaches [23] detect natural clusters (groups) from the dark based on statistical features only. In this case, giving a label to each extracted cluster could be a challenging task. Consequently, semi-supervised methods are used in [71, 22, 26, 27, 30, 31] consisting of combining a mixture of labelled and unlabeled data as training input. A large set of ML algorithms are tuned to provide high performance. The most commonly used algorithms from the non-supervised branch are k-means and k-nearest neighbor. The category of supervised MLA includes Hidden Markov Models, Adaboost, Ripper, Support Vector Machines (SVM), C4.5 and Naive Bayes. As our approach is based on predefined output classes, we focus on supervised category algorithms. Several works [27, 25] compare these algorithms performance. It turns out that C4.5 decision tree algorithm [79] performs better in terms of accuracy and computational speed.

Table 2.3 A summary table of cited papers and used classification methods properties

Reference	Publication year	Input features		Machine learning method			Output Granularity		Early identification	Encryption awareness	Dataset			Retraining consideration	Deployability consideration
				Method category											
		Flow level	Packet level	Sup.	Non/semi Sup.	Algorithm	Fine	Coarse			Artificial	Real	Reliable Ground-truth		
[71]	2006	✓	✓	✓	✓	HMM, k-nearest neighbor		✓		✓		✓			
[22]	2007		✓		✓	Gaussian Mixture Model		✓	✓	✓		✓			
[23]	2009	✓			✓	K-means, genetic Programming		✓				✓			
[24]	2009	✓	✓	✓		Ripper, C4.5		✓			✓	✓			
[25]	2009	✓		✓		Ripper, SVM, Adaboost, C4.5, Naïve Bayes		✓				✓			
[26]	2010		✓		✓	k-nearest neighbor, k-means		✓	✓			✓			
[27]	2011	✓	✓	✓	✓	Naïve Bayes, Multi-Layer Perception, Ripper, Random Tree, k-nearest neighbor, J48		✓	✓	✓		✓			
[28]	2011	✓		✓		MOGA, k-means, C4.5		✓				✓			
[29]	2011	✓		✓		Naïve Bayes		✓		✓	✓		✓		
[30]	2011	✓			✓	Hierarchical k-means		✓				✓			
[73]	2012	✓		✓		C5.0		✓		✓	✓		✓		
[31]	2013	✓	✓		✓	k-means		✓			✓		✓		
[72]	2014		✓	✓		C4.5, SVM		✓	✓	✓	✓		✓		
[74]	2015	✓		✓		Hoeffding Adaptive Tree	✓			✓		✓	✓	✓	
Our work	**2017**		✓	✓		**C5.0**	✓		✓	✓		✓	✓	✓	✓

2.6.6 Retraining Considerations

Only few proposals address formally the retraining process which is required to ensure the temporal robustness of a classifier. In fact, such issue is addressed only in [74] over the surveyed works using Hoeffding Adaptive Tree (dynamic update of the tree branches). Such lack in the literature limited the deployment of MLA for traffic classification.

2.6.7 Deployability Considerations

Among the studied approaches, deployability of the proposed solution is never considered. In this thesis, we address such question by implementing our proposed solution on a real Home Gateway prototype. Moreover, both accuracy and resource consumption are evaluated using real-traffic scenarios. While the first issue is a concern addressed by all proposed approaches, the second one is mistakenly ignored.

2.6.8 Discussions and Positionning

As illustrated in Table 2.3, several lacks are reported while studying the literature. Our first observation is related to the used dataset. Most of the presented studies were based on an unreliable dataset. In fact, we consider a dataset as reliable if it is real (ISP or campus) and labelized using a reliable ground-truth generation process. Hence, only the dataset used by authors in [74] fulfills such criterias. In Chapter 4, we address these lacks and present a large-scale analysis of residential traffic collected on real ISP network. At this aim, we detail our data collection and ground-truth generation methodology.

Secondly, early classification which is mandatory to perform real-time management actions is addressed only in [22, 26, 27, 72]. Thirdly, early proposed approaches deal with encrypted traffic but do not provide a fine-grained classification output. Furthermore, retraining and deployability considerations have never been addressed despite their importance regarding the viability of a classicfication approach. In Chapter 5, we propose an early, fine-grained classification approach based on the C5.0 machine learning algorithm. Moreover, we validate its deployability while integrated as part of our Home Gateway based probe. A retraining architecture is also designed to ensure the vialbility of our proposed scheme.

2.7 Towards Autonomic Home Network Management

Exported IPFIX records are collected on the collector side to allow plan and to execute management and optimization rules. Thus, our logic is to construct an autonomic control loop which automates home network management process. In this thesis, we assume that our architecture is based on MAPE-K (Monitor Analyze Plan Execute - Knowledge) paradigm [82]. MAPE-K control loop was first introduced by IBM for Autonomic computing. It consists of a closed feedback loop that could be summarized as follows in our Home Network context:

- **Touchpoints:** Touchpoint are composed by a set of sensors and effectors. On the one hand, sensors expose information about the current state of a managed resource. On the other hand, effectors allow the change of a state of a managed resource. In our context, the Home Gateway and the managed HNIDS are seen as-sensors when monitoring active flows and become effectors when actions must be performed (i.e. Path selection, traffic blocking, etc.)
- **Monitor (HNMC side):** Collects the details from the managed resources (i.e. topology information, configuration property settings, etc.). The monitor function aggregates, correlates and filters the collected details until it determines a symptom that needs to be analyzed.
- **Analyze (HNMC side):** Based on the symptom provided by the monitor function, deep data analysis and reasoning (i.e. machine learning model) is performed. If changes are required, a change request is passed to the Plan function.
- **Plan (HNMC side):** Based on the change request provided by the analyze function, actions are structured to achieve a desired alteration on managed resources. Change plan can take many forms, ranging from a simple command to a complex workflow. The obtained change plan is logically passed to the Execute function.
- **Execute:** As a final step, the Execute function is responsible of changing the behavior of managed resources using effectors (Home Gateway and managed HNIDS).
- **Knowledge:** The knowledge is a set of shared data among the Monitor, Analyze, Plan and Execute functions. Updates may occur if the Monitor function raises unobserved information.

2.8 Conclusion

In this chapter, we exposed strengths and weaknesses of existing architectural approaches for Home Network monitoring. We focused on flow monitoring architecture and present possible bottlenecks that need to be evaluated in our Home Network context and that will be addressed in Chapter 3. Then, we presented an analysis of existing application identification approaches. Our interest was turned to MLA based approaches. Several lacks have been identified in the literature. Dataset collection and analysis were pointed as a crucial step in such approaches and will be addressed in Chapter 4. Finally, we propose a novel scheme of early residential traffic classification based on C5.0 machine learning algorithm. Our scheme is implemented and validated as part of a Home Gateway exporter in Chapter 5.

Chapter 3 Traffic Monitoring in Home Networks: Enhancing Diagnosis and Performance Tracking

3.1 Introduction

In this thesis, we focus on home network traffic monitoring which involves several tasks including active flows identification, per flow performance tracking and services matching. In fact, enabling services monitoring is a major requirement to enhance existing topology monitoring systems [10] as it offers multiple benefits. For end-users, it provides a better knowledge of their home network usage (devices with high bandwidth consumption, running applications, etc.). For ISPs, it provides enablers to improve QoS as highlighted in [81]; it would also allow applying advanced parental control, anomaly detection [11], etc.

In this chapter, we study the feasibility of ISP devices (Home Gateway, HNIDs such as PLC plugs and Wi-Fi extenders) based flow monitoring approach. These devices have typically constrained resources (CPU and memory) inducing technical challenges for traffic monitoring. After depicting our proposed architecture in Section 3.2, we select a promising tool to evaluate its performances using both real and synthetic traffic and upon a representative testbed (section 3.3). The aim of our evaluation is to detect possible deployment issues and bottlenecks. Finally, we conclude the chapter (Section 3.4).

3.2 Flow Monitoring in Home Networks: Our Architectural Approach

In this section, we propose to study the feasibility of ISP devices based flow monitoring approach. Our monitoring approach involves: active flow detection, real-time flow performance monitoring and application identification (each flow is associated with its generating application). Achieving these tasks produces a home network traffic detailed knowledge. Based on this collected knowledge, autonomic management and optimization tasks could be performed efficiently.

Figure 3.2 shows the designed overall architecture to perform these tasks. Three distributed major components are detailed:

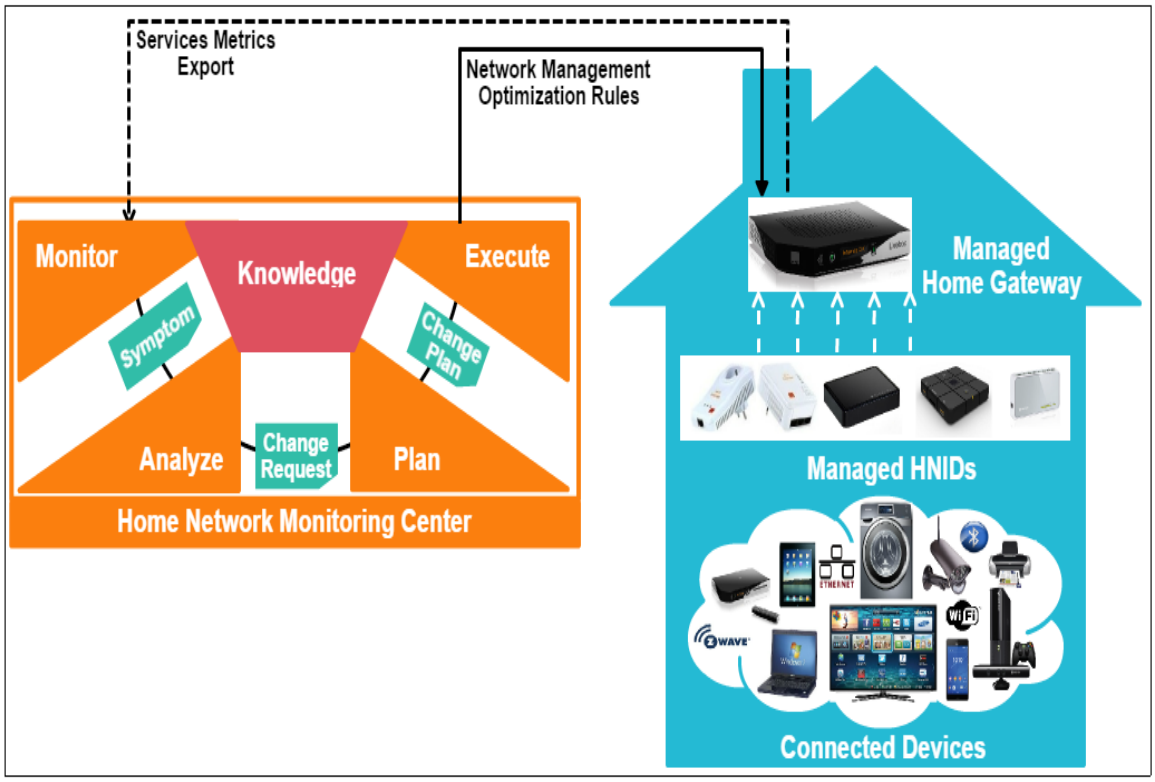

Figure 3.1 Home network monitoring approach architecture

- **Managed Home Gateway,** which refers to a flow monitoring enabled Home Gateway. While traffic passing through the Home Gateway is directly monitored, some intra LAN traffic may not pass through the Home Gateway (e.g. a file transfer between 2 devices connected to an Ethernet switch). Therefore, it raises the need to

extend the observation process to secondary sensors. In such case, a proxy feature may be enabled to forward HNID's records providing a unified export.

- **Managed HNIDs**, which refer to a flow monitoring enabled HNIDs (i.e. Wi-Fi extenders, PLC plugs). Enabling these devices as additional sensors aims to provide a full visibility of home network usage. Moreover, enabling adequate filters (Intra LAN traffic monitoring only) will avoid data redundancy on Home Gateway.
- **Home Network Monitoring Center (HNMC)** is in charge of collecting and analyzing exported Home Gateway records. Combining analysis results with topology information would allow advanced home network diagnosis and troubleshooting. A first deployment option for the HNMC is on a home network local device (typically the Home Gateway). A second option could be to perform the data collection in the Cloud from several home networks. The first option will guarantee continuous monitoring compared to a cloud placement (e.g. when there is a WAN access issue). On the other hand, the second option is more suited to avoid supplementary burden on the home gateway (constrained resources) and would allow more advanced analysis based on several households' data. Both options support remote access by the support desk (after user agreement).

3.3 nProbe tool experimental Evaluation

3.3.1 nProbe as a suitable tool?

Open source IPFIX tools are developed both for IPFIX probes (exporters) and collectors. As we aim to assess possible limitations and bottlenecks for resource constrained devices, we focus on exporter's proposed tools. We consider only exporters implementing at least one of the needed features for our approach (performance metrics and application identification) as detailed in Table 3.1.

Considering our approach needs, we select nProbe as the promising exporter to test. As an "all in one" option, this tool offers the largest set of configuration's possibilities. Moreover, it integrates native packet capture optimization techniques. Furthermore, integrated in various low-resource devices (raspberry Pi, SFP), nProbe proved its embeddability (binary size < 100 KB).

Table 3.1 Comparative summary of IPFIX exporters

	Version	Packet capture library	Performance metrics	Application Identification	Sampling option
YaF[1]	2.7.1	libpcap		DPI (regular expression matching for 46 protocols)	
nProbe[2]	7.1	Libpcap, PF_RING/ZC	Application and network latency, TCP metrics, plugins	nDPI (170 protocols)	Packet, Flow
pmacct[3]	1.5.1	libpcap		Optional patch (L7-filter library, RTP, eDonkey)	Packet, Flow
QoF[4]	0.9.0	libtrace	TCP metrics		

In this section, we study the performances of the nProbe exporter running on an experimental testbed. It is a first step evaluation conducted on laptops devices. The target is to estimate the load induced by such tools before testing them on resource limited devices (home gateway) as a second step. As nProbe was designed and validated for high speed networks (10 Gbps), our evaluation is not focusing on packet processing speed but rather on resources consumption (CPU usage, memory usage, bandwidth load) of the exporter device. Furthermore, we selected ntopng [14] which is an open source collector provided within the same project.

3.3.2 Testbed setup

Our testbed is designed to fit major home network configuration use cases (multiple devices/OS and connectivity technologies) as depicted in Figure 3.2. As nProbe is not integrated into our Home Gateway (CPU: 500 MHz, Memory: 128 MB), we used port mirroring configuration to simulate gateway based packet capture. Therefore, we run the nProbe tool on a Linux PC receiving all the traffic sent to the home gateway. Moreover, we compiled nProbe with both standard libpcap and PF_RING libraries. Indeed, testing both libraries allows us to evaluate the benefits of integrating fast processing techniques in our approach.

[1] https://tools.netsa.cert.org/yaf/
[2] http://www.ntop.org/products/nprobe/
[3] http://www.pmacct.net/
[4] http://www.ict-mplane.eu/public/qof

Finally, per process resource consumption computation is performed by the atop/netatop tool. We tested both real and synthetic traffic scenarios as detailed in the Figure 3.4.

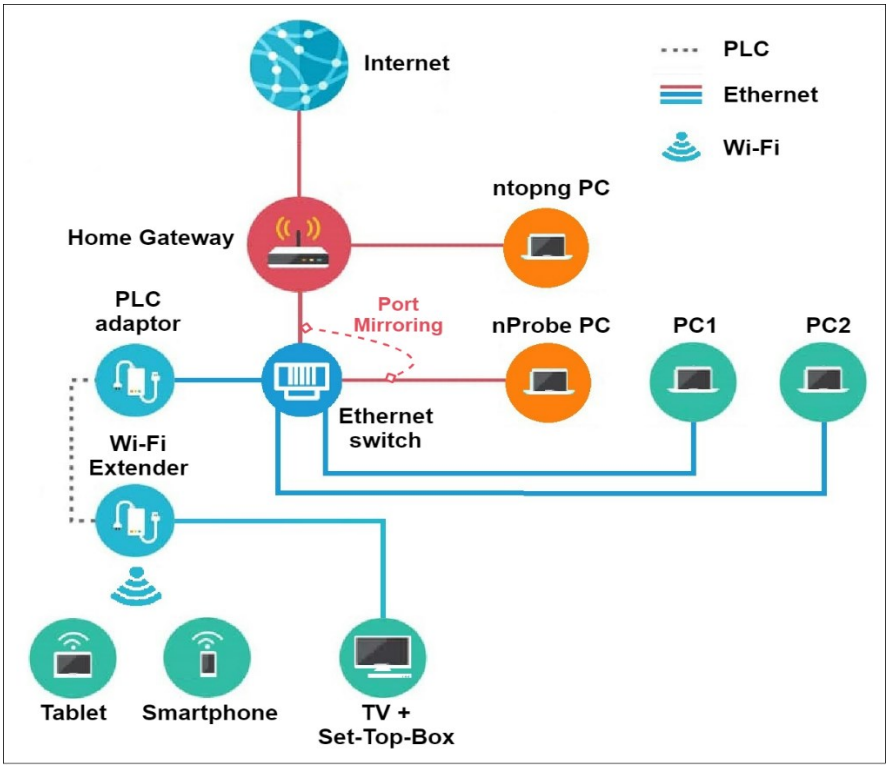

Figure 3.2 Testbed configuration

We disabled the Wi-Fi access point on the Home Gateway (Livebox 3) and simulate the same behavior using a PLC/Wifi extender plugs. Such configuration allows us to generate real traffic using multiple wireless devices (Smartphone, Tablet). Real traffic scenario includes also TV traffic generated using a STB connected to the Ethernet port of the Wi-Fi extender.

3.3.2.1 Real traffic scenario

We emulate a user's typical heavy load scenario as depicted in Figure 3.4 with 34.75 Mbps average traffic throughput (5135 unique flows with an average IP packet size of 1046 bytes) while capturing a full packet trace. Then, we replay the captured traffic using *Pfsend* traffic generator (PC1) to evaluate several nProbe configurations:

- LPCAP-DPI: nProbe compiled using libpcap and nDPI enabled
- LPCAP-NODPI: nProbe compiled using libpcap and nDPI disabled
- PFRING-DPI: nProbe compiled using PF_RING and nDPI enabled
- PFRING-NODPI: nProbe compiled using PF_RING and nDPI disabled

Also, one should note that such evaluation configuration allows us to evaluate the generated bandwidth load by the exporter in real life scenarios.

3.3.2.2 Synthetic traffic scenario

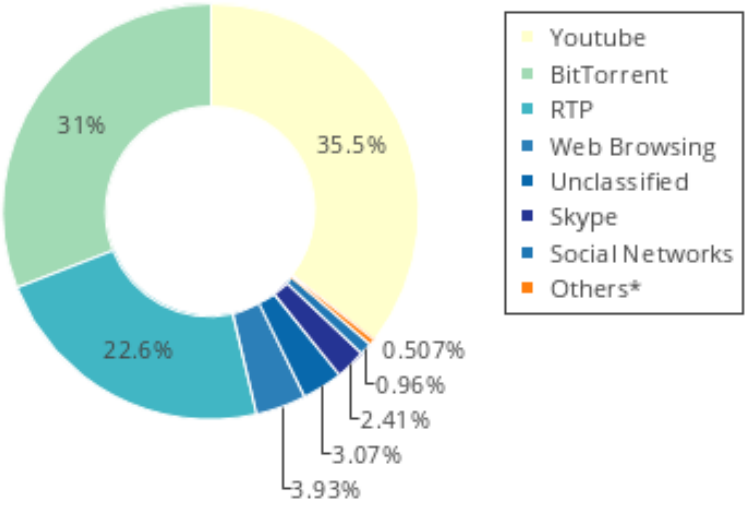

Figure 3.3 Collected real traffic distribution (obtained by nDPI)

We modified our testbed removing all devices except of PCs. We run a traffic generator and a traffic receiver on PC1 and PC2, respectively (port mirroring configuration is set up between PC1 and nProbe). Then we generate, in a first series of tests, UDP synthetic traffic (packet size of 1500 bytes) at several rates (50 Mbps, 200Mbps, 400Mbps, 600Mbps, and 800Mbps). A second series of tests consist of setting the rate of UDP traffic at 200 Mbps while varying the number of parallel flows. While the aim of the first series of tests is to evaluate CPU and memory usage under several data rates, the latter focuses on the impact of the number of entries in the flow cache.

Indeed, we emphasize that varying the number of parallel flows has the same effect of varying the number of hosts as it is processed on the same way on the flow cache.

3.3.3 Performance evaluation results

At the probe side, we evaluate the resource consumption in terms on CPU usage and memory. Furthermore, we evaluate additional network load generated by the exporting process.

3.3.3.1 CPU Usage

Measured CPU usage is observed on one 2.6 GHz core processor. A first observation is that nProbe-PFRING configuration runs on an idle network with an average of 10% CPU usage as shown in Figure 3.5(a). This is mainly due to a technical implementation choice. Indeed, nProbe's developers implemented an active polling approach instead of the classical passive one (use of *usleep* until packets arrive) to ensure an optimal packet capture in high speed networks scenarios. This explains why nProbe-LPCAP (passive polling) provides a better performance while traffic load is less than ~450 Mbps. A second important observation is that CPU usage is almost constant with respect to the number of flows (with a constant total rate of 200Mbps) except for very low number of flows (less than 25) as shown in Figure 3.5(b). We suppose that it is mainly due to the flow cache management process which is not optimized for a low number of entries. A final observation is concerning the DPI impact on device's CPU usage as shown in Figure 3.4(a). The observed average loads for LIBPCAP-noDPI/DPI and PFRING-noDPI/DPI are 5.01%, 6.16%, 13.7% and 15% respectively.
We conclude that enabling application identification on real traffic costs 9.3% average additional overhead comparing to nProbe-PFRING disabled DPI scenario. A higher overhead is observed using standard libpcap where it reaches up to 23%. However, the main outcome is that the activation of nDPI causes a relatively low CPU load in both cases.

3.3.3.2 Memory Usage

In our approach, the memory usage is expected as the most probable bottleneck. In fact, while CPU capacity is evolving according to the Moore law, memory resource evolves in a slower way. We focus on real traffic scenario to evaluate possible bottlenecks. While memory usage is low and stable: less than 12 MB, enabling DPI increases memory load up to 32MB in our test scenario as illustrated in Figure 3.4(b). The increase follows a continuous trend which might be an issue when running the application identification for a long period of time. This is mainly explained by the nDPI data caching feature. Fixing the maximum allocated cache size according to the device memory capacity might allow limiting this increase.

3.3.3.3 Network Load

Our last evaluation focuses on the bandwidth load generated by the exported data. In the real traffic scenario (with DPI enabled), we observe an average load of 156 Kbps with spikes up to 6.67 Mbps corresponding to export instants as depicted in Figure 3.4(c).

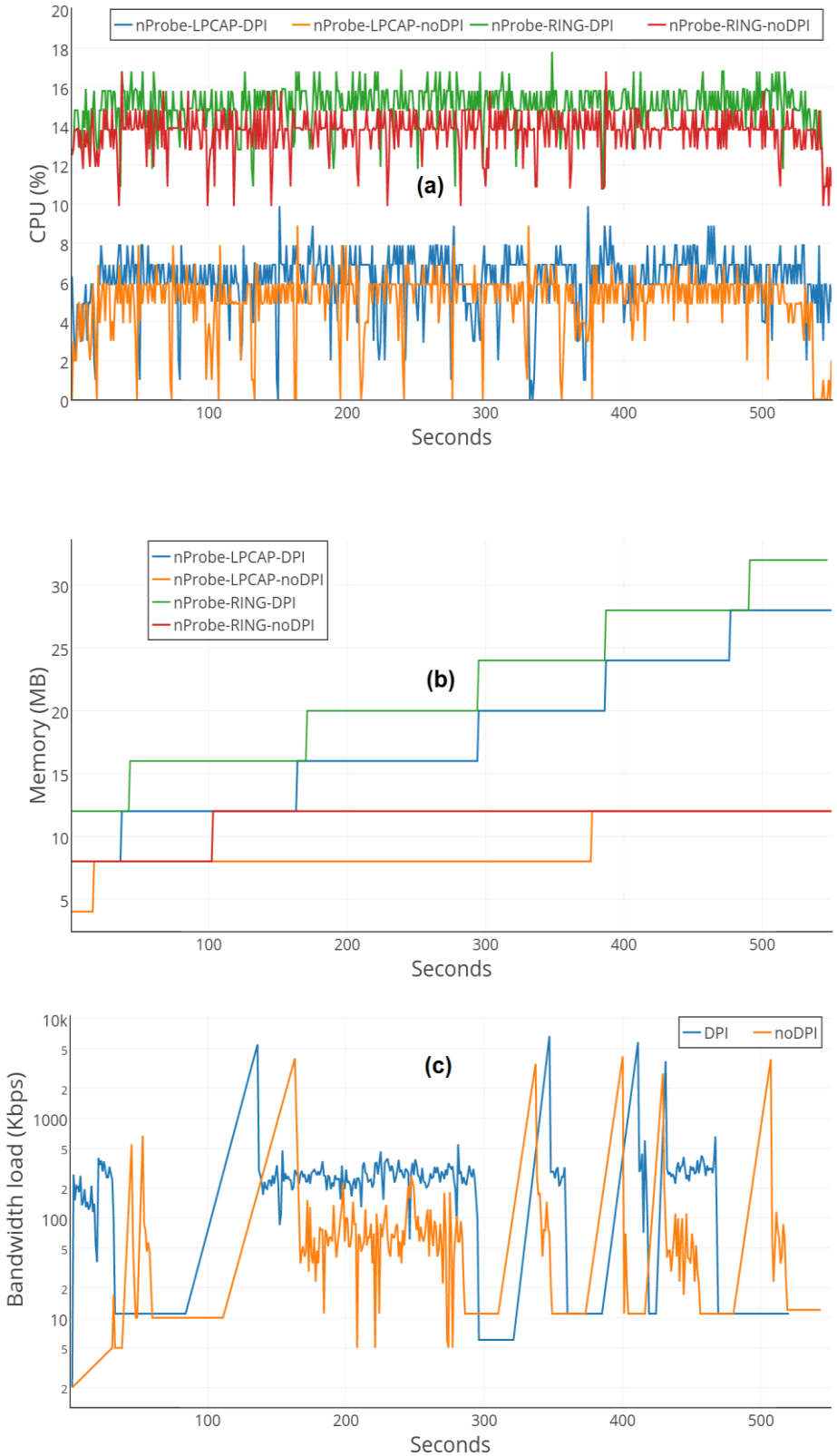

Figure 3.4 nProbe resource consumption using real traffic scenario

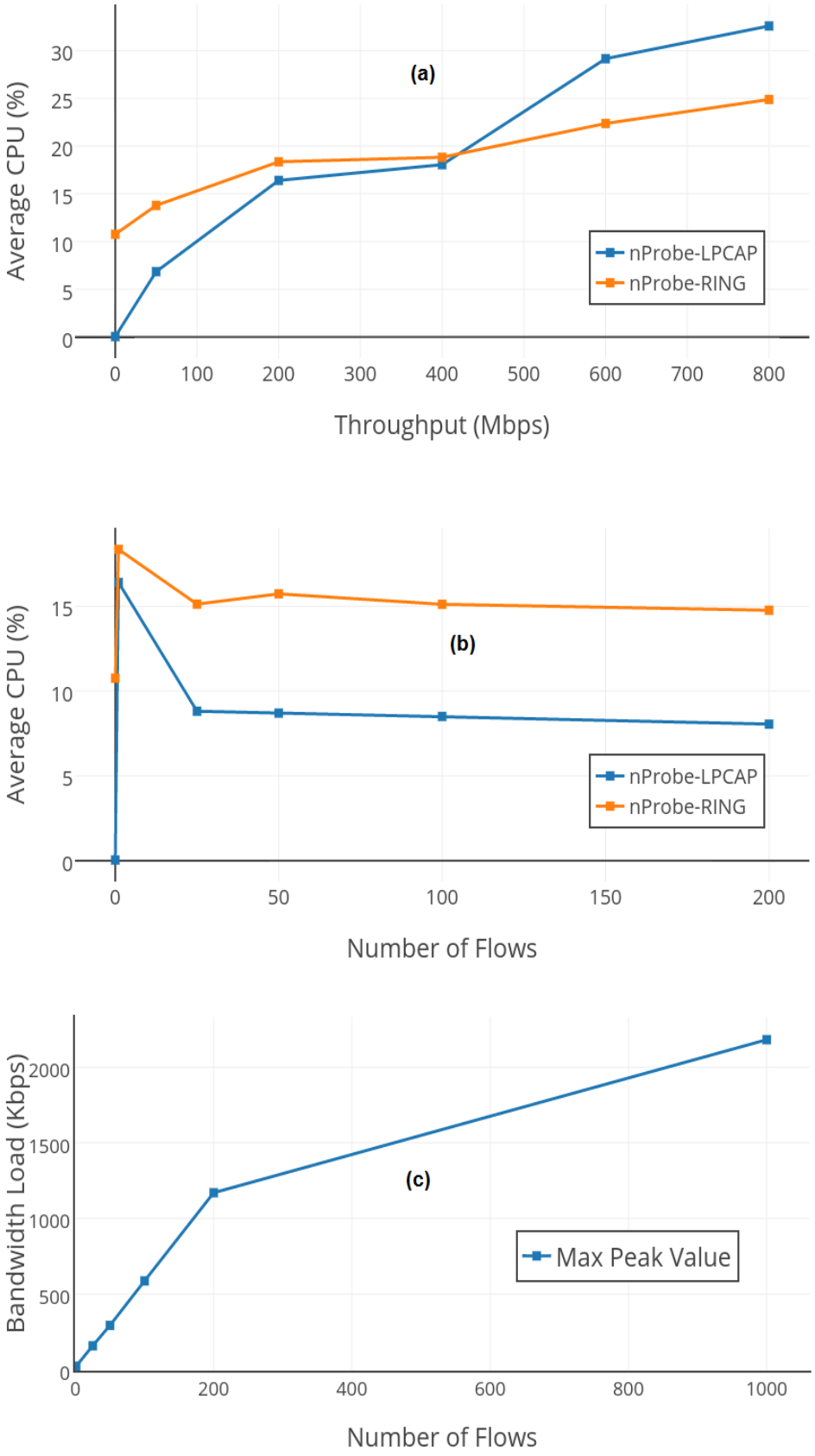

Figure 3.5 nProbe resource consumption using synthetic traffic scenarios

As the network load depends on various parameters (e.g. flows duration and/or number of flows) which we do not control in the real traffic scenario, we decided to focus on maximum peak rates while varying the number of flows. Therefore, we used long lived (300 seconds) synthetic controlled UDP flows with constant rate of 200 Mbps. The metering process is mainly driven by active timeout expiration entry which determines the frequency of exporting data (120 s default value). As depicted in Figure 3.5(c), throughput spikes maximum value increases while varying the number of generated flows: 1.17 Mbps for 200 parallel flows, 2.6 Mbps for 1000 flows case. In our approach, we must keep the overhead as low as possible as access link upstream rate might be a bottleneck. Fortunately, major IPFIX exporters allow tuning entry expirations timeouts and rules which allows controlling efficiently exported data rate.

3.3.3.4 Synthesis and discussion

To sum up, we can say that nProbe performance, under several scenarios, are satisfactory. Indeed, L7 monitoring using a standard library (libpcap) capturing heavy load real traffic needs less than 6% of the average CPU usage on a single 2.6Ghz CPU core. Moreover, memory average load is less than 20 MB. Finally, IPFIX exporting induces an average overhead of 156Kbps with the conducted scenarios.

We expect near future home gateways to be less resource constrained (e.g. Dual core processor with at least a 512MB RAM) which would facilitate the deployment of such traffic monitoring tools as advocated in section 3.2. However, almost all modern gateways integrate hardware packet processing accelerators (no packet visibility at kernel-level) leveraging packet capture process to a new challenging level. We will address this issue in Chapter 5.

Finally, to illustrate the benefits of traffic monitoring for improving home network diagnosis tools, we used ntopng [14] to provide traffic statistics information to the Home Network Assistant tool proposed in [2]. As shown in Figure 3.6, combining both topology and flows monitoring information provides better insight on the home network (bandwidth usage repartition per device, ongoing flows on each links, etc.). The aim is to rely on such information for troubleshooting purposes in a self-care mode (done by the customer) and customer care fashion (done by the ISP hotline). Indeed, it would help to reduce the ISP hotline costs.

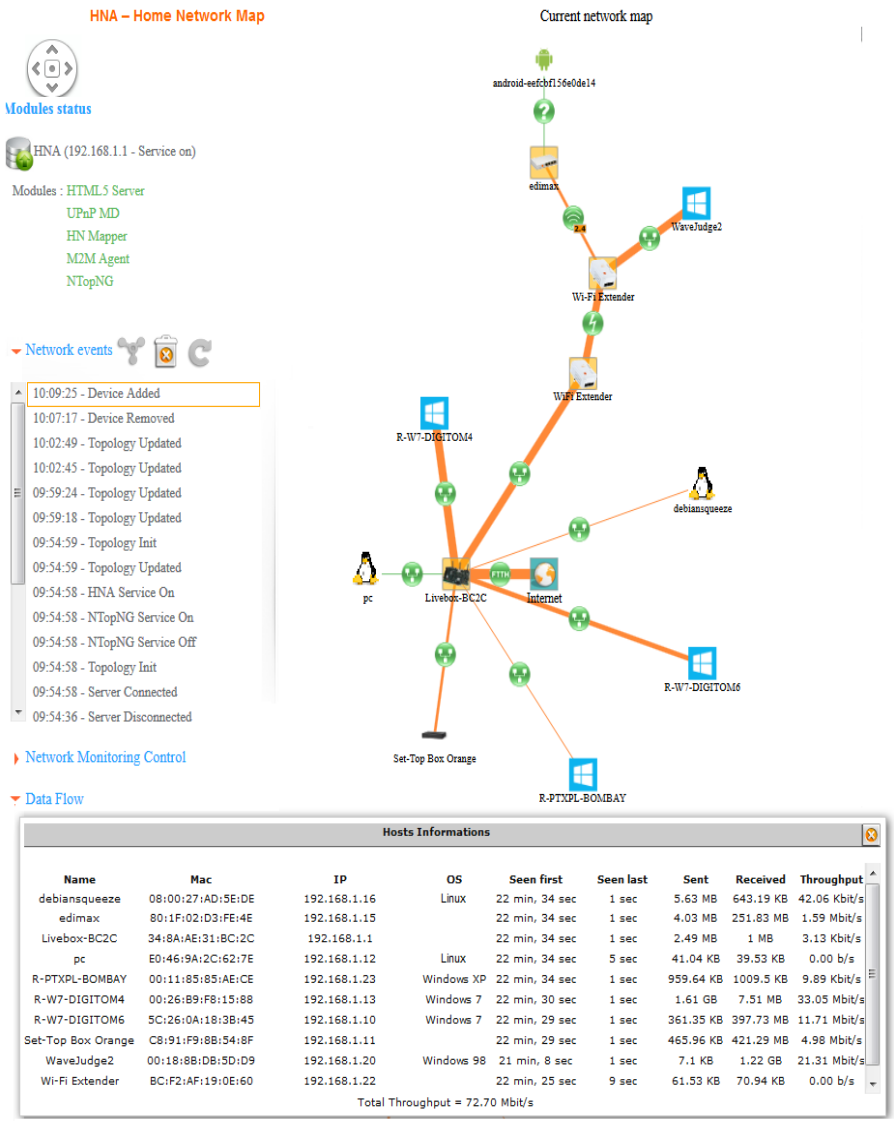

Figure 3.6 HNA traffic monitoring screenshot

3.4 Conclusion

Home networks services monitoring is a key feature to improve users Quality of Experience (QoE) and reduce ISP hotline costs. In this chapter, we provided an architectural approach to perform enriched home network monitoring. Then, we evaluated the nProbe tool on an experimental testbed focusing on resource consumption criteria.

While the obtained results are promising, our work highlighted several possible challenges and improvements. Our conducted experiments were based on Deep Packet Inspection and provided useful insights related to the overhead induced by such technology indicating its feasibility.

However, we believe that such technique is not viable due the stated reasons in chapter 2 (signature engineering, traffic encryption, etc.). To design and deploy efficiently our proposed architecture components, we focus on building a reliable MLA approach. At this aim, a deep understanding of home networks traffic and usages must be achieved as it is presented in the next chapter. Indeed, the trace captures presented in the next chapter are the input data set for the training of our machine learning algorithm described in Chapter 5.

Chapter 4 Towards Understanding Residential Networks' Usages: From Packets to Customers

4.1 Introduction

The continuous change of home networks usages is raising several challenges. From an ISP standing point, managing efficiently the home network portion, yields to a better customer's satisfaction while reducing help-desk costs. As stated in Chapter 3, real time traffic characterization in home networks faces several challenges. In fact, one crucial step when addressing this task is a deep understanding of users' modern Internet usage. Reporting traffic patterns and usages profiles helps to understand the demands of today and the challenges of tomorrow. Thus, optimization components, such as application aware QoS controller [81] or content caching optimizer [83], could be efficiently designed and deployed.

A large body of the literature [84, 85, 86, 87, 88] presented fixed access characterization focusing on application usages. Four major observations can be reported while studying the state of the art.

The first observation is concerning the measurement points. In fact, a major part of the observations is reported from IP backbone level [84]. This implies collecting both business and residential customers' traffic despite the significant gap that might exist between both profiles.

The second observation is regarding the traffic classes' granularity. In most of the papers [86, 89], traffic analysis granularity is too coarse to tackle the growth of web applications. In fact, traffic classes such as Web, P2P, DNS, SSH, etc. were sufficient in the early days of the

Internet where web activity was limited to visiting some text and images contents hosted by a savvy set of servers. Nowadays, web applications are growing in a tremendous fashion resulting on tens to hundreds of embedded objects loaded from several dedicated servers such as media streaming portals, online gaming platforms or social network servers. Moreover, some major over-the-top (OTT) actors such as Google are starting to deploy their own delivery protocols such as QUIC (Quick UDP Internet Connections). Last but not least, virtual tunneling technologies (including Tor) are evolving rapidly and are more adopted by residential users due to several factors (e.g. privacy concerns, European Union Intellectual Property Rights Enforcement Directive (IPRED) and HADOPI law in France). The above described evolutions require the need of a fine-grained classification while reporting traffic patterns and profiles.

The third observation is concerning the reliability of traffic identification engines and is closely related to the previous observation. While some researchers used coarse grained open source identification engines [87], some works achieve fine grained classification provided by commercial tools [85, 88, 5] with unknown precision performances. This weakness affects the reliability of the reported results and is inherited from the traffic classification field where the community warned about the lack of common benchmark standards [54].

Our last observation is concerning the lack of subjective studies of home networks usages and residential customers' behavioral habits. In fact, previous works are conducted objectively based on packet traces only. A subjective view tackles customer's habits and usages trends that cannot be unveiled using network data and is mandatory to achieve a complete view of residential networks usages.

In this chapter, we aim to overcome these lacks. To do so, we first present residential users' traffic characterization and usage pattern identification based on real traces collected at the closest points to the users. Moreover, our fine-grained analysis is partly based on an open source engine with well-known precision performances. Then, we rely on our own developed tools to analyze the traces more in depth. By doing so, we are able to focus on service categories of most popular applications. Finally, we conduct a subjective behavioral analysis of more than 600 residential customers through a questionnaire. The obtained knowledge is used to perform a complete synthesis of residential network usages. To the best of our knowledge, this is the first time that such detailed analysis of residential Internet usages, combining the above-mentioned criteria, is presented.

The rest of the chapter is organized as follows. In section 4.2, we present some relevant works used as landmarks during the design of our network data collection and processing methodology which is given in Section 4.3. Then, in section 4.4, we depict residential users' traffic characteristics based on our collected data set at several scales focusing on applications usages level. In section 4.5, we present our subjective analysis of residential networks usages. Then, we discuss the obtained results in comparison with other reports in Section 4.6. Finally, we conclude the chapter (Section 4.7).

4.2 Related Work

Traffic characterization reported works are distributed among several countries over continents. We already mentioned few of them in the previous section highlighting the lacks that we would like to overcome. Internet traffic usages differ considerably between countries [5] and, thus, in this section, we consider only measurements performed on the same French ISP's broadband access network. Note that reported dates in the rest of the chapter refer to dataset capture dates.

In early 2006, Siekkinen et al [89] reported the low bandwidth utilization measured at 1300 Asymmetric Digital Subscriber Line (ADSL) users scale, mainly explained by P2P applications upload limited rate from "producer" side. More than half of the traffic was unidentified due to the use in this study of a port-based traffic classifier.

The early fiber access deployment (2008, July) impact is studied in [87] among 1182 ADSL and 1905 FTTH (Fiber To The Home) customers, respectively. Authors reported that a large part of upstream traffic was unclassified and, thus considered only TCP flows with SYN packet observed (to improve classifier accuracy) while characterizing the traffic applications distribution. The described breakdown confirmed that P2P applications were the main consumers of uplink capacity whereas the downlink bandwidth was mainly used by video streaming.

In [85], a recent traffic characterization (2013, October) from two major European ISPs is reported. The French observed customers' pool size was 7500 with the third identified as FTTH users. While unclassified traffic ratio was not reported in this study, video streaming reaches up to 36% of downstream classified FTTH customers' traffic (26% for ADSL) followed by P2P applications (16% and 12%, respectively). Uplink traffic was dominated by P2P applications (78% and 48%, respectively). Authors reported that for FTTH access uplink (resp. downlink), 3% (resp. 15%) of customers generates 80% of the measured traffic.

4.3 Network Data Collection and Processing

As introduced in the previous sections, several works providing traffic behavior characterization do not provide their measurement's logic. Based on "Tell me how you measure me, and I will tell you how I will behave" principle, we depict in the following our measurements processes.

4.3.1 Network Data Collection

Our network data collection process is performed at a major French ISP residential aggregation network. Two measurements servers are located between the clients and a Broadband Remote Access Server (BRAS) as depicted in Figure 4.1. This configuration allows capturing bidirectional flows generated by residential customers as well as small and medium enterprises. Moreover, managed services such as IPTV and VOIP are excluded at this observation level. Consequently, the traffic characterization presented in this chapter differs from previous contributions which were based either on backbone or an academic network traffic.

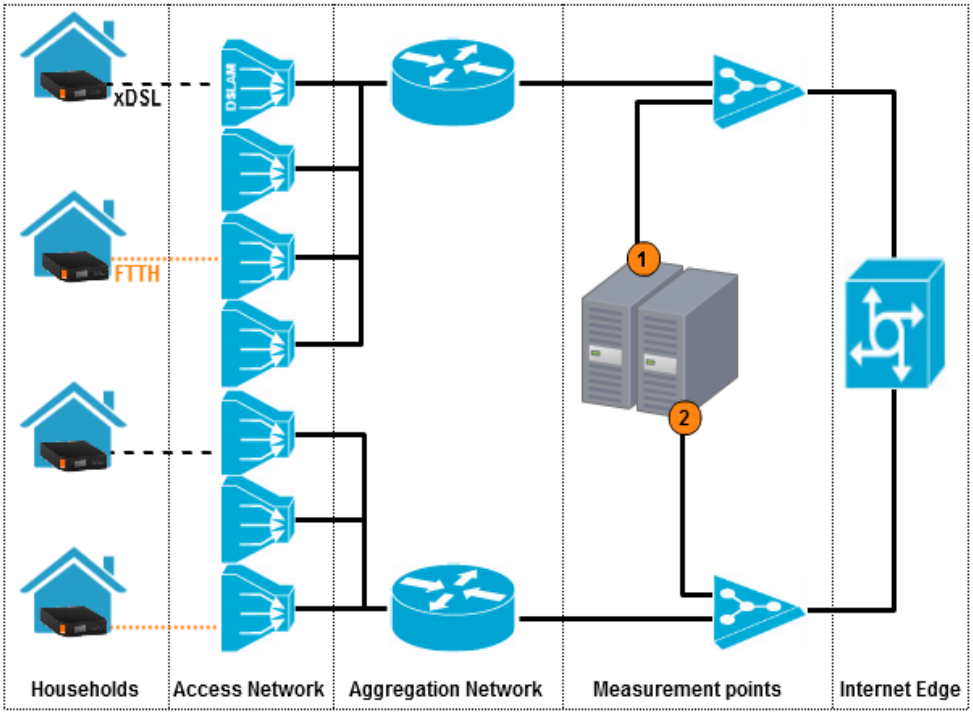

Figure 4.1 Overview of the collection process architecture

Based on the daily behavior of residential traffic reported in the literature [85, 88], two traffic packet traces are captured on July 8th, 2015 at 12am and November 26th, 2015 at 8pm, respectively. While the first one (referred as JUL8-12) is obtained using server 1, the latter

(referred as NOV26-8) is obtained using both servers and it captures a commonly identified "busy hours" period. Note that dates spacing is realized to provide more than a "one shot" view of the traffic. Finally, all data are anonymized as per the French laws on personal integrity.

Table 4.1 Details of collected traffic traces

	Duration	Clients					Flows						Volume					Packets
		Total	FTTH (%)	ADSL (%)	VDSL (%)	Other (%)	Total	FTTH (%)	ADSL (%)	VDSL (%)	Other (%)		Total	FTTH (%)	ADSL (%)	VDSL (%)	Other (%)	
JUL8-12	33 min 56 sec	20,889	88.5	4.4	0.7	6.4	5,104,680	83.3	5	1.4	10.3	up	23,593.95	85	1.3	1	12.7	117,400,351
												down	76,153.7	89.9	3.1	1.4	5.6	
NOV26-8	1 H 59 min 53 sec	43,184	50.3	42.6	5.8	1.3	50,455,020	47.3	45.2	6.4	1.2	up	117,913.27	75.6	17.9	5.6	0.9	741,989,892
												down	648,348.6	55	37.1	6.9	1	

The collected data is combined with per customer additional knowledge such as subscribed offer, uplink (resp. downlink) negotiated rate between the Home Gateway and the access node (Digital Subscriber Line Access Multiplexer (DSLAM) or Optical Line Termination (OLT)), etc. Thus, it allows us to have a detailed view of our collected traces as described in Table 4.1. Moreover, enterprise customers' data are filtered to focus on residential subscribers only. Few customers are not represented (experimental lines and in progress termination subscribers) below. Note that, "Other" category refers to residential subscribers that do not use provided the ISP Home Gateway. We assume that "Other" class is mainly dominated by enterprise customers that subscribe to residential offers for saving money purposes while including some few "geek" customers. Our hypothesis is consolidated by the breakdown between the two traces. Despite being collected on both servers, "Other" customers' traffic aggregated volume in NOV26-8 trace is 7 times lower than the one observed in JUL8-12 trace. This is mainly explained by the time slice of JUL8-12 trace which is a working hour. Consequently, "Other" customers' class traffic is excluded in our reported results. While this issue is not addressed in the literature, we consider that such class of customers' may affects

the reported results as upstream traffic volume generated reaches up to 12.7% of the overall traffic in working hours.

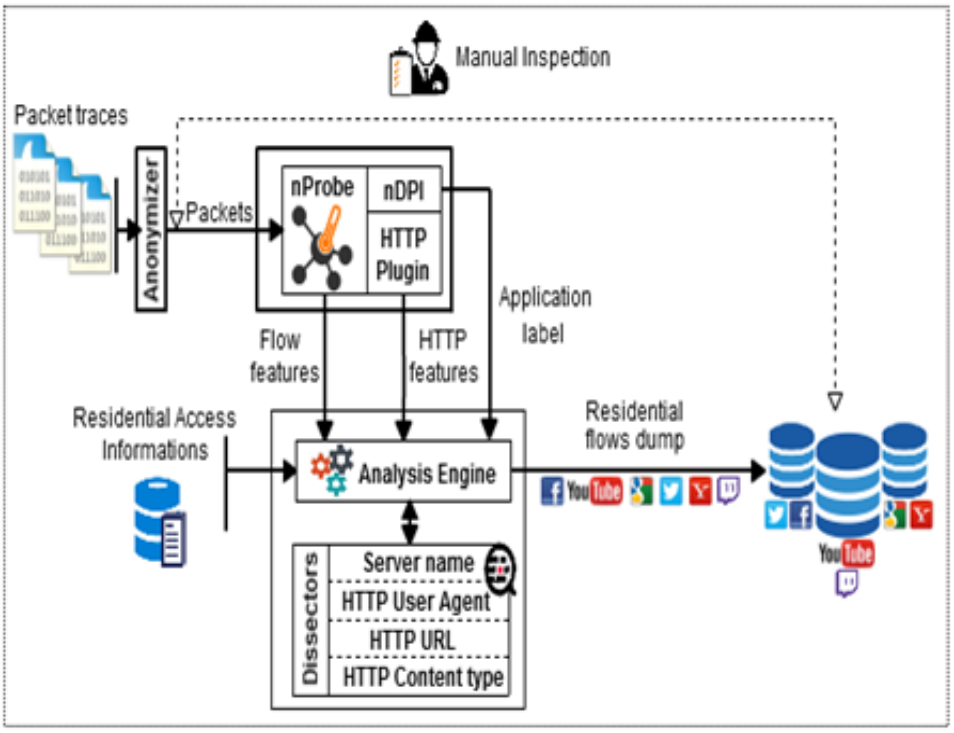

Figure 4.2 Data processing overview architecture

4.3.2 Network Data Processing

Packet-level traces allow the aggregation of traffic at several abstraction scales. In the following, we identify flows as bidirectional connections based on the classical 5-tuple {protocol ID, source IP address, destination IP address, source port, destination port}. Flow expiration is mainly driven by inactive timeout (120 sec) or natural expiration (FIN packet observed for TCP flows). Concerning the flow direction definition, we consider that data transmitted from the customer's Home Gateway to the measurement point is counted as "uplink" while the reverse direction is called "downlink".

Flow features are extracted using the nProbe [25] tool; nProbe (v7.3) allows the extraction of the main flows' characteristics in addition to some advanced performance metrics[5]. Besides, we enabled HTTP plugin which performs HTTP headers extraction (i.e. URL, content type, user agent, etc.). Consequently, HTTP flows entries are enriched with their corresponding headers.

[5] http://www.ntop.org/nprobe/how-to-monitor-latency-using-nprobenagios-world-conference-europe/

The application identification engine is partly performed using the nDPI library [15]. As mentioned above, one common weakness in some fine-grained traffic analysis works is the reliability of used commercial classifiers [85, 88, 5]. The detailed dissection logic implemented in nDPI is open source. Its main core consists of a combination of payload patterns matching (Aho-Corasick algorithm) and IP address mapping approach while port-based identification is used as a last resort. Accuracy performance of nDPI is evaluated with respect to other both open source and commercial engines in [68]. Authors reported the high accuracy of nDPI which outperformed the other evaluated tools. While authors used nDPI version 1.6, we used in this work the latest available version (1.7) which introduced several improvements such as the QUIC protocol dissector among 223 applications set.

Moreover, we developed a set of parsers to refine the classification results. In fact, the used nDPI version allows defining a set of consistent application labels such as {protocol.sub-protocol} (e.g. HTTP.Facebook). We enrich sub-protocol identification for HTTP flows using knowledge extraction from HTTP headers (HTTP User Agent, URL and content type). In addition, flows' sub-protocols identification is also affined based on resolved server name parsing. Based on these inputs (user agent, url, reverse dns), a majority vore strategy is applied. A manual labelization is performed in case no majority was established. Such approach allows us to provide in-depth view of application classes such as Video on Demand, Live TV, Advertising, etc. which is not provided by nDPI. The several tasks involved while processing capture traces (depicted in Figure 4.2) results in a reliable fine-grained analysis.

Table 4.2 Explanation of traffic categories

Traffic categories	Examples
Real-Time entertainment	Video streaming sites (YouTube, NetFlix, Twitch, etc.), Streamed or buffered Audio or video content (RTP, RTMP, Flash, MPEG), Video Advertising, etc.
Gaming	Steam, WorldOfWarcraft, LeagueOfLegends, CandyCrush, Scrabble, Console portals, etc.
Social Networking	Facebook, Instagram, Google+, Tinder, LinkedIn, Viadeo, Tumblr, Pinterest, Twitter, etc.
Storage	FTP, DropBox, AppleiCloud, UpToBox, 1fichier, CrashPlan, etc.
Marketplaces	Software Update/Download, GooglePlay, AppleAppStore, AppleiTunes, Windows Update, etc.
Administration	DNS, MDNS, NTP, NetFlow, STUN, etc.
Web Browsing	Google, Advertising, Yahoo, News Portals, LeBonCoin, HTTP_Others, etc.
Tunneling	HTTP_Proxy, SSL_Others, Tor, SSH, IPSec, etc.
Filesharing	P2P (Bittorrent, eDonkey, Gnutella), Newsgroups, etc.
Communications	Skype, Viber, YahooMail, Gmail, WhatsApp, IMAPS, POP3, etc.
Others	GoogleNow, GoogleMaps, QUIC, etc.

Finally, applications are categorized into main service categories as shown in Table 4.2. Our definition of each category is based on the one provided in [5]. The categorization is performed in a protocol ascendant way. For example, "DNS.YouTube" is categorized as Administration services while "HTTP.YouTube" is categorized as Real-Time entertainment. Thus, applications analysis is performed at three consistent granularities scales which are the application layer (L7) protocol (e.g. HTTP), the application itself (e.g. YouTube) and the traffic categories (e.g. social networking).

4.4 Traffic Analysis and Characteristics

We describe in this section the traffic characteristics extracted from the captured traces. Our analysis is presented as a walk through the TCP/IP protocol stack layers.

4.4.1 Overview of the Aggregated Traffic

The aggregated traffic measurements provide useful insights to grasp the general traffic patterns in residential networks.

The average number of flows in progress observed is 27562 (resp. 79253) for the JUL8-12 (resp. NOV26-8) trace with an average aggregate rate of 345.6 Mbps (resp. 810 Mbps) as depicted in Figure 4.3. While no major fluctuation is reported for JUL8-12 trace, we observe a decreasing activity for [8pm-9pm] time slot in NOV26-8 trace. This could be explained by typical dinner time slot (French main news broadcast time) correspondence.

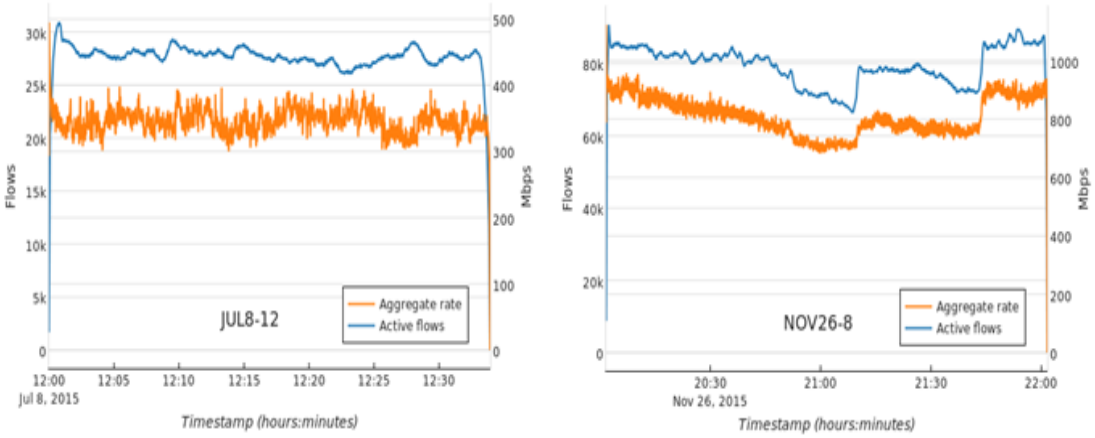

Figure 4.3 Number of flows in progress and aggregate rate

Note that we do not consider one packet flows to compute the number of flows in progress. Indeed, the number of flows having only one packet represents 44% (resp. 53%) of the observed flows in JUL8-12 trace (resp. NOV26-8) as depicted in Figure 4.4.a. However,

single packet flows constitute only 0.2 % (resp. 0.4%) of overall data volume in JUL8-12 trace (resp. NOV26-8). This observation is also reflected on flows duration (see Figure 4.4.b) where the average flow duration is 21 secs (resp. 24 sec) for flows counting at least two packets.

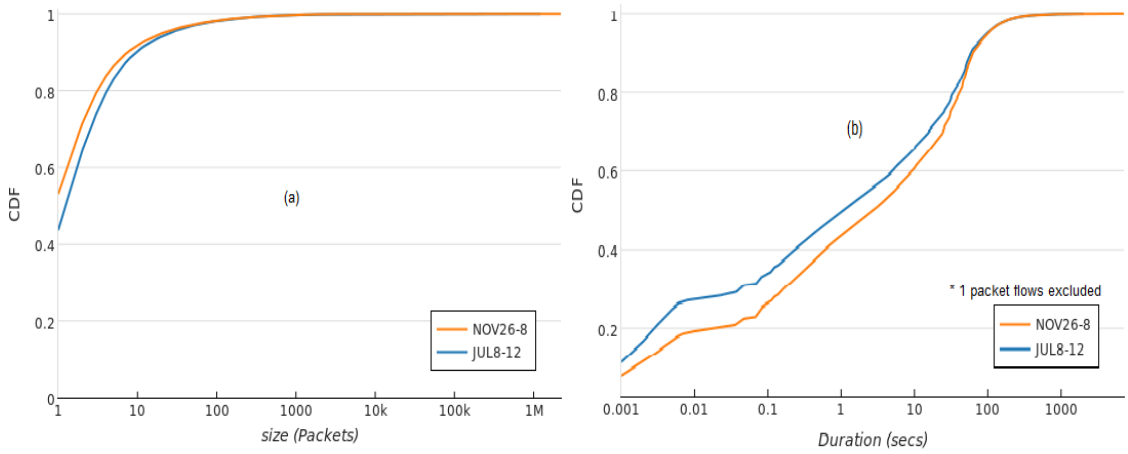

Figure 4.4 Cumulative distribution function of flows length

Figure 4.5 shows the proportions of flows contributing to the cumulated total data volume. It highlights that 1% of flows generate 80% of the observed data volume on both traces. As expected, most flows are short, but few long flows contribute to most of the total volume.

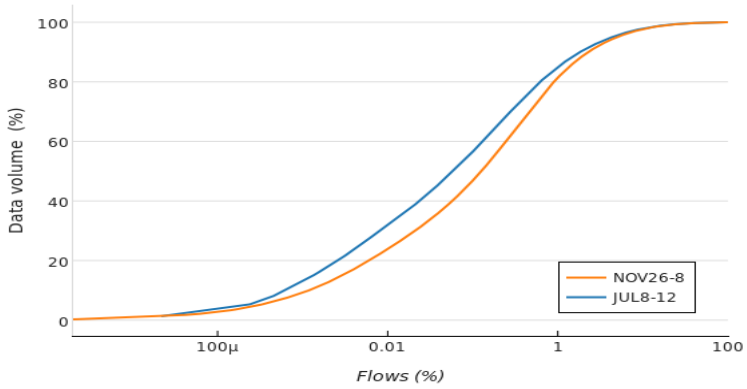

Figure 4.5 Flows contribution to transferred data volume proportions

Table 4.3 Packet size repartition

	Total	Size clusters				
		Up to 128	128 to 256	256 to 512	512 to 1024	124 to 1500
JUL8-12	108,539,550	36.13%	3.46%	2.05%	2.4%	55.96%
NOV26-8	915,820,653	38.42%	3.26%	1.73%	1.54%	55.05%

Finally, an analysis of packet sizes reveals that most packets have either a size around the MTU or are very short (less than 128 bytes: TCP SYN and ACK typically) as detailed in Table 4.3.

4.4.2 Costumers' Behavior Analysis

Table 4.1 summarizes the data volume generated by each access type in both upstream and downstream directions. While the maximum line speed is 1Gbps (resp. 200Mbps) in the downlink (resp. uplink) for FTTH customers, the maximum downlink (resp. uplink) negotiated rate for xDSL customers is 102 Mbps (resp. 26Mbps). Note that JUL8-12 trace data volume is mainly generated by FTTH customers due to the used capture configuration (only server 1 enabled). On the other hand, despite that the number of FTTH customers is almost the same as xDSL customers for NOV26-8 trace, the volume generated in upstream (resp. downstream) by FTTH customers is 3 times (resp. 1.25) higher than the volume generated by xDSL customers. As expected, since FTTH access provides higher rates, the corresponding users tend to generate more traffic. Moreover, xDSL customers' upstream channel limitation is observed in the ratio between the downstream volume and the upstream one. In fact, the computed ratio in JUL8-12 (resp. NOV26-8) trace is equal to 6.2 (resp. 10.3). The traffic is more balanced for FTTH customers with a ratio equals to 3.4 (resp. 4) due to the increased capacity of the upstream channel in the FTTH case.

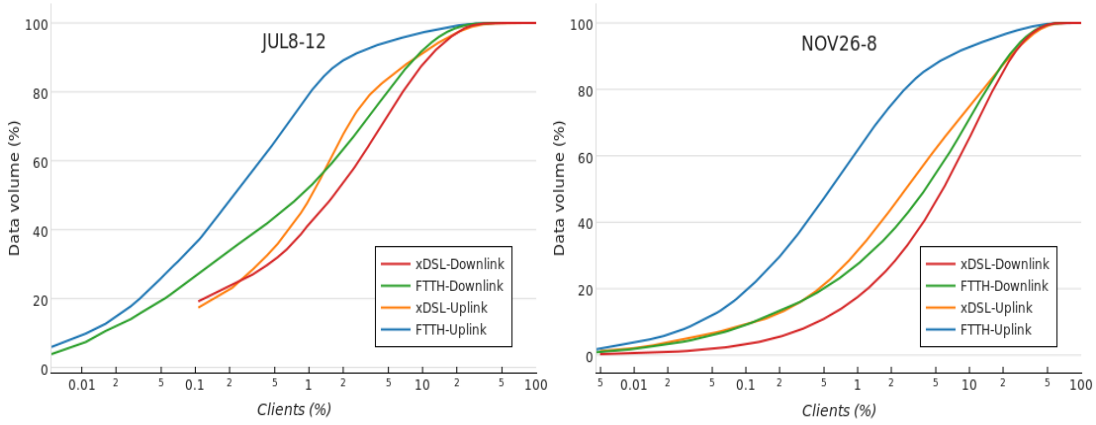

Figure 4.6 Customers contributions to transferred data proportions

One major observation while analyzing customers' behavior in both traces is the impact of "heavy users" on the overall generated traffic. In fact, 80% of the overall data volume in JUL8-12 (resp. NOV26-8) trace is generated by 5% (resp. 15%) of the customers. To provide an in-depth analysis, we clustered household's traffic according to their access types and to the channel direction as illustrated in Figure 4.6.

Almost 80% of the FTTH upstream traffic is generated by 1% (resp. 2.7%) of customers in JUL8-12 trace (resp. NOV26-8) while 5 times more customers are generating 80% of the downstream traffic. In the xDSL networks case, 80% of the upstream traffic is generated by 3.6% (resp. 13.2%) of customers while 7% (resp. 16.5%) of them generates the same ratio in the downstream direction. These results indicate that the impact of heavy users is more dominant in the upstream direction than in the downstream for both xDSL and FTTH access types. It may suggest that heavy users generate types of applications requiring more traffic on the uplink with respect to the other users (e.g. P2P applications, cloud storage, etc.). In addition, 6% (resp. 10%) of FTTH heavy users' population in JUL8-12 trace (resp. NOV 26-8) is the same on both upstream and downstream directions meaning that 94% (resp. 90%) of heavy users' customers' pool are considered as heavy users only on one direction.

In a second step, we focus on the average link utilization of the observed customers. The results are summarized in Figure 4.7 and show a low average bandwidth usage per customer. As we can see, the maximum observed link utilization per customer in JUL8-12 trace is 6.7%, while it is 12.3% for NOV26-8 trace.

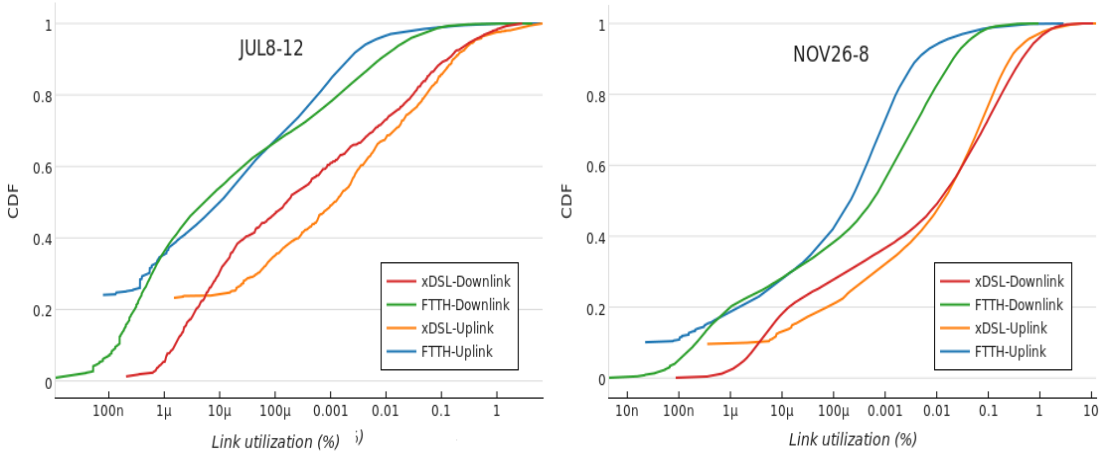

Figure 4.7 Cumulative distribution function of per customer average link utilization

4.4.3 Transport Layer Characteristics

Table 4.4 summarizes the transport protocol statistics observed in both traces. Unsurprisingly, TCP connections represent 92.6 % (resp. 89.6%) of the downstream traffic volume in JUL8-12 (resp. NOV26-8), while UDP traffic is low. We observe an increasing ratio of UDP traffic volume in the uplink direction up to 34% in JUL8-12 (resp. NOV26-8) (resp.45.3%). This proportion change is mainly due to P2P application as explained in the next section in addition to the fact that TCP traffic in the upstream direction usually involves ACK packets.

Finally, the UDP/TCP bytes ratio which is 0.15 (resp. 0.18) in JUL8-12 (resp. NOV26-8) trace is greater than the ratios reported in previous studies [90]. While researchers expected UDP expansion to be mainly driven by some UDP based P2P applications (e.g. uTP, PPLive), we assume that this could be reinforced by new OTT delivery protocols such as QUIC as we will see in the next section.

Table 4.4 Transport layer protocols distribution

	Flows		Volume (Bytes)					
	TCP (%)	UDP (%)	TCP (%)			UDP (%)		
			up	down	total	up	down	total
JUL8-12	68.1	31.1	65.9	92.6	86.7	34	7.2	13.1
NOV26-8	69.4	30.1	54.2	89.6	84.1	45.3	10	15.4

Table 4.5 Higher layer protocols statistics

			HTTP	SSL	Unknown	P2P	QUIC	RTP, RTMP	GRE, IPSec	IMAP, IMAPS, POP2, POPS	DNS, MDNS	Others
Flows (%)		**JUL8-12**	26	25.2	31.8	6.2	1.3	0.06	0.4	1.5	4.8	2.7
		NOV26-8	29.6	27.8	29.3	4.5	1.1	0.04	0.2	0.9	4.7	1.8
Volume (Bytes)	**Downlink**	**JUL8-12**	45.6	37.1	10.4	1.5	2	0.8	0.46	1.6	0.03	0.5
		NOV26-8	49.6	32.4	9.6	2.3	3	1.2	0.9	0.3	0.07	0.6
	Uplink	**JUL8-12**	5.63	21.7	57.6	11	0.7	0.3	0.5	0.4	0.05	2.1
		NOV26-8	10.2	16.4	56.4	12.1	1.1	0.4	0.8	0.5	0.3	1.8
	Total	**JUL8-12**	36.7	33.7	20.9	3.5	1.7	0.7	0.47	1.36	0.04	0.93
		NOV26-8	43.5	29.9	16.8	3.8	2.7	1.1	0.8	0.3	0.1	1

4.4.4 Higher Layer characteristics

Application layer protocols statistics in terms of flows and bytes are depicted in Table 4.5. As can be seen, HTTP and SSL (including HTTPS) traffic dominates, with 70.4% (resp. 73.4%) of total traffic volume in JUL8-12 (resp. NOV26-8) trace. Finally, early QUIC protocol deployment is also observed in both traces contributing up to 2% (resp. 3%). We assume that the observed increase between the traces is mainly due to Google deployment strategy adopted in 2015 [91]. P2P traffic represents less than 4% of the total volume, although it is higher on the uplink side. The contribution of the other protocols is quite low, except for the unknown part.

4.4.4.1 Digging into the unknown

20.9% (resp. 16.8%) of total traffic volumes are classified by nDPI as unknown. Our initial hypothesis is that 'unknown' traffic is dominated by encrypted P2P traffic. Our hypothesis is consolidated by the asymmetric distribution of "unknown" class contributing up to 57% (resp. 56%) of uplink traffic. Moreover, we isolated the 2,556,142 unique distant peers involved in unknown flows in both traces. The resolved peers' names inspection reveals that the isolated pool is mainly dominated by ISP customers (more than 80%). Thus, P2P protocols constitute most of the unknown class. Consequently, the impact of the French HADOPI law is reflected through the observed dominance of obfuscation in P2P protocols.

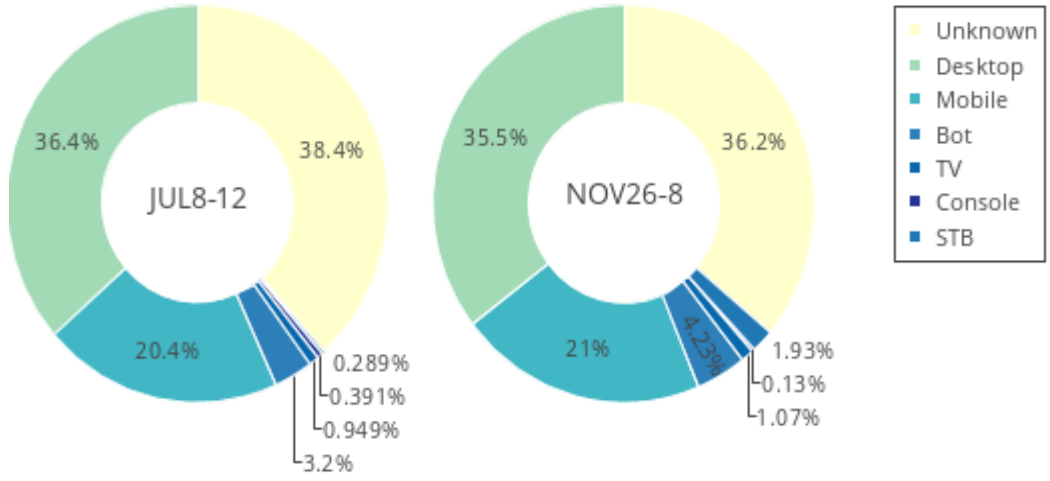

Figure 4.8 Device type traffic breakdown

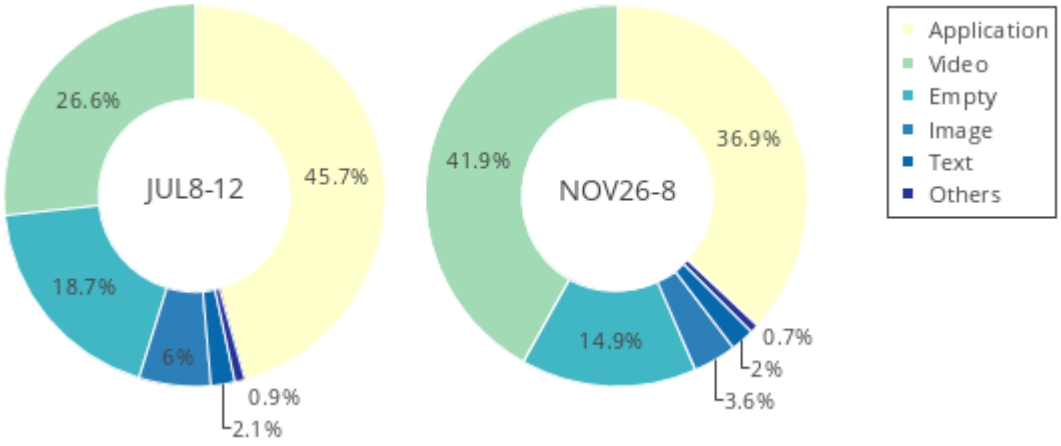

Figure 4.9 Content type traffic breakdown

4.4.4.2 HTTP traffic characterization

We focus on HTTP traffic characteristics to understand more in depth the main dominant protocol observed in both traces. Based on the extracted User Agent field in each flow, we build a device type detector able to recognize more than 80% of extracted user agents. In addition to well-known existing parsers[6][7] , we developed a third parser and rely on a majority voting output. Manual inspection is performed when three distinct outputs are obtained. Traffic volume distribution among detected device types is depicted in Figure 4.8. Empty user agents and some bot-like user agents that we are unable to identify are referred as 'Unknown'.

As a second step, we focus on content types used by HTTP servers to indicate the type of the content they are delivering to the clients. As depicted in Figure 4.9, application type is the most prominent in JUL8-12 trace generating up to 45% of the total traffic volume, followed by video[8] contents. This trend is reversed in NOV26-8 trace where the video dominates representing 43% of the downlink traffic volume. While 58% of HTTP transfers do not indicate a content type value in JUL8-12 (resp. NOV26-8) trace, more than 70% of those transfers are generated by desktop devices. It suggests that desktop HTTP applications are subject to less careful design than mobile ones.

Table 4.6 Traffic categories statistics

			Real-Time entertainment	Gaming	Social Networking	Storage	Marketplaces	Administration	Web Browsing	Tunneling	Filesharing	Communications	Others
Flows (%)		**JUL8-12**	1.9	0.3	2.6	0.6	0.3	6	33.4	8	36.9	3.8	6.2
		NOV26-8	2.6	0.3	4.3	0.4	0.2	5.4	31.7	11.6	32.8	4	6.7
Volume (Bytes)	**Downlink**	**JUL8-12**	25.3	3.6	3.1	2.6	1	0.04	32.4	13.9	11.7	2	4.36
		NOV26-8	36.5	3.6	3.6	2.1	1	0.1	29.5	5.9	11.4	1.5	4.8
	Uplink	**JUL8-12**	1.9	0.2	0.8	1.8	0.1	0.08	17	6.7	68	1.5	1.9
		NOV26-8	4.1	0.5	1.5	0.7	0.2	0.5	10.8	6.2	66.8	4	4.7
	Total	**JUL8-12**	20.1	2.9	2.6	2.4	0.8	0.1	29	12.3	24.2	2	3.6
		NOV26-8	31.5	3.1	3.3	1.9	1	0.2	26.6	6	19.9	1.9	4.6

[6] https://www.npmjs.com/package/device

[7] https://pypi.python.org/pypi/user-agents

[8] We include other content types referring to video content such as "application/vnd.apple.mpegurl"

4.4.5 Traffic Services Analysis

Traffic categories statistics are depicted in Table 4.6. A first observation is concerning Filesharing traffic which dominates in both traces in terms of flows and upstream traffic volume (more than 67%). While a large part of Filesharing traffic is obfuscated, Bittorrent is the main identified application used. Note that unknown traffic which we detect as P2P as explained in section 4.4.4.1 fall into this category.

Web browsing dominates the traffic volume in JUL8-12. Our category definition covers both HTTP (almost 80%) and SSL traffic that do not fall into other defined categories. While we acknowledge that more investigation is needed, we assume that this category may cover some different traffic categories that we are unable to detect. As an example to illustrate this issue, encrypted traffic generated by a subset of Content Delivery Networks (CDNs) that we classify as Web Browsing may fall into the real-time entertainment category.

Real-time entertainment traffic contributes up to 20% of the total traffic volume in JUL8-12 and dominates NOV26-8 trace (31%). Video streaming applications are the most prominent, in particular, YouTube as detailed in Table 4.7.

Tunnelling class represents 12.3% (resp. 6%) of the traffic volume in JUL8-12 (resp. NOV26-8) trace. The generated volume is dominated by SSL_unresolved traffic; it refers to identified SSL transfers to servers that do not provide reverse DNS response and where we are unable to extract a readable certificate name. While Tor applications usually use such SSL exchanges, we assume that other services could also use such configuration.

Facebook largely dominates the Social Networking category (90%) which contributes up to 3.6% of downlink traffic volume in NOV26-8 trace.

Gaming traffic generates up to 3% of the total volume in both traces. Traffic generated by the Steam platform represents more than 50% of the generated volume. Note that in NOV26-8 trace, Candy Crush generates up to 22% of observed gaming flows.

DropBox dominates storage traffic in terms of flows while 1fichier.co is the most prominent in terms of volume.

Regarding marketplaces traffic which is equal to almost 1% in both traces, software update flows leads the category in term of flows in both traces. In terms of volume, the trend is dominated by both Windows Update and Apple_iTunes.

The communication category represents up to 2% of the observed traffic volume on both traces. IMAPS contributes up to 62.2% of the traffic volume in JUL8-12 trace while Skype

leads the category in terms of flows. In addition, Skype generated 68% of communications traffic volume observed in NOV26-8. Note that "others" category is mainly dominated by QUIC flows and Google (1e100)[9] flows that we are unable to categorize. Finally, the administration traffic is mainly composed of DNS traffic.

Table 4.7 Zoom on per category applications

		JUL8-12		NOV26-8	
		Flows (%)	***Bytes (%)***	***Flows (%)***	***Bytes (%)***
Real-Time entertainment	**YouTube**	65.5	54.3	68.1	46.5
	NetFlix	4	15.7	3.9	16.8
	LiveTV	5.2	8.7	11.2	20.3
	VoD	2.7	4.9	2.2	5
	Twitch	3	4.6	2.7	2.8
Gaming	**Steam**	42.5	52.9	45.5	66
	Candy Crush	9	0.01	21.9	0.05
	Console Portal	8.9	4.9	6.4	12.5
Social Networking	**Facebook**	89.1	93.1	93.7	89.8
	Twitter	4.6	1.6	1.9	1
	Instagram	3.8	3.6	2.9	7.8
Storage	**DropBox**	52	32.1	43.3	2.4
	AppleiCloud	15.3	0.3	27.8	0.7
	1Fichier.co	0.3	32.7	1.1	73.8
Marketplaces	**Apple_iTunes**	36	65.3	38.4	30
	WindowsUpdate	11.7	17.4	10.9	57.3
	Software update	47.8	8.1	45.3	3
Tunnelling	**SSL_unresolved**	57.9	82.6	35.2	71.8
	HTTP_Proxy	1.2	2.2	42.9	5.6
	Tor	4.4	8.7	0.2	0.2
Communications	**Skype**	45.9	12.9	67.6	67.7
	IMAPS	30.1	62.2	17.4	12.6
	Viber	6.8	1.5	6.7	2.4

A last observation is concerning previously identified heavy users. More precisely, we focus on the traffic distribution of such cluster of customers. In the upstream, heavy users' traffic volume is mainly dominated by Filesharing flows up to (73%). In the downstream, the main part of the traffic is composed of real-time entertainment, Filesharing applications and Tunneling traffic. Our conclusion is that FTTH access links capacity allows a subset of customers to act as P2P seeds which turns them into heavy users.

4.5 Subjective Analysis of Home Network Usages

In this section, we present the results of our conducted subjective analysis of home network usages. The purpose of the study is to correlate the knowledge extracted from the packets

[9] https://support.google.com/faqs/answer/174717?hl=en

traces with subjective truth obtained from customers to build a complete view of residential network usages.

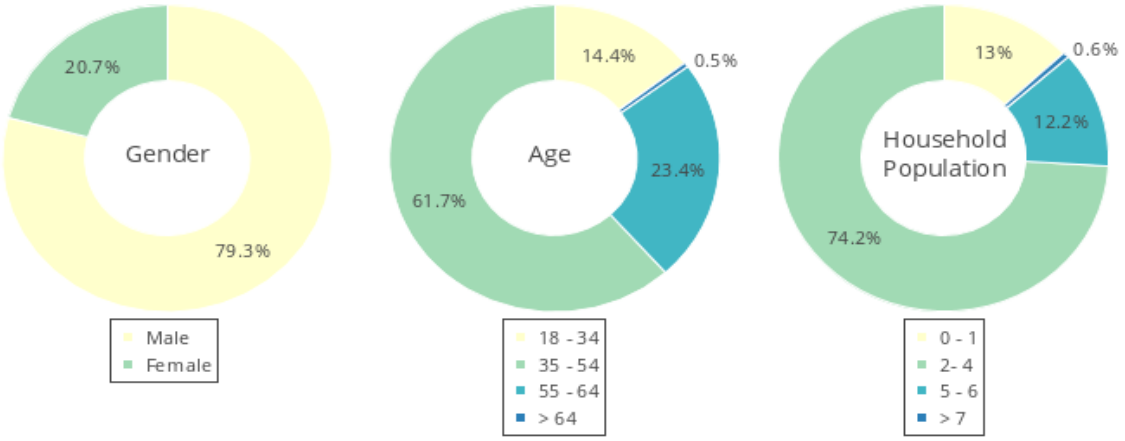

Figure 4.10 Interviewed population characteristics

4.5.1 Overview of the study

We prepared an online questionnaire addresses sent to users via Email. The subjects are invited to answer a set of questions to unveil their behavioral habits while connected to the Internet through their fixed access. The study is conducted among 645 subjects during both October and November 2015 (same period of the packets traces capture). The interviewed population characteristics are depicted in Figure 4.10. Furthermore, interviewed customers are in France and are part of the employees of the same ISP involved in the packet traces capture. While the interview questions are detailed in Appendice A, our logic could be summarized as follow:

- We study home networks complexity by focusing on the home network topology of interviewed customers (number of connected devices, distribution among several devices types, etc.)
- For each Internet service category, we focus on the popularity of widely used applications.
- For each Internet service category, we depict usage distribution among connected devices in the home network. Such information is not available through network data analysis. As mentioned in Section 4.4.4.2, device type is extracted using HTTP User Agent only for HTTP flows and thus, could not provide sufficient knowledge regarding other protocol usages.

One of the aims of this study is to identify potential correlation between devices types and the associated traffic types (e.g. social networks usage on smartphone/tablet rather than desktops, etc.).

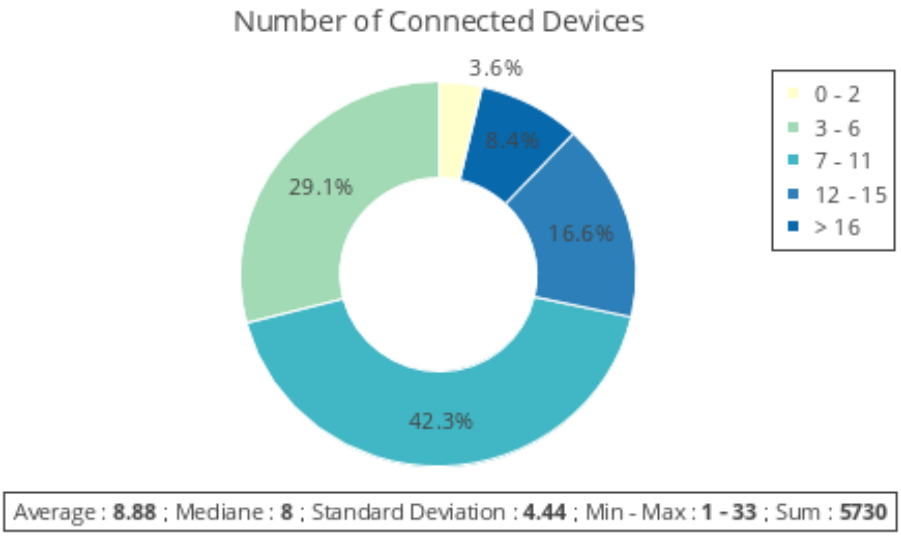

Figure 4.11 Number of connected devices per household

Table 4.8 Per Household connected devices distribution

	Average	Standard deviation	Median	Min -Max	Sum
PCs	2.45	1.34	2	0 - 9	1579
Smartphones	2.35	1.25	2	0 - 7	1517
Tablets	1.05	0.91	1	0 - 5	680
Set-Top-Boxes	0.83	0.52	1	0 - 3	537
Gaming Consoles	0.53	0.75	0	0 - 4	340
Connected TVs	0.46	0.61	0	0 - 3	296
Embedded Cards (i.e. Raspberry Pi)	0.22	0.75	0	0 - 9	144
Network Attached Storage (NAS)	0.35	0.53	0	0 - 3	224
Connected Radio Stations	0.24	0.67	0	0 - 7	153
Connected Audio Amplifiers	0.18	0.54	0	0 - 6	117
Others	0.22	0.73	0	0 - 9	143

4.5.2 Residential Networks Topology

In this section, we focus on the topology of the subjects' residential networks. Our obtained results confirm the impact of the proliferation of users end-devices observed during the last decade. In fact, a typical home network is composed by an average of 9 connected devices as depicted in Figure 4.11. Moreover, 67.3% of interviewed persons had more than 7 connected devices.

In a second step, we focus on the composition of the identified set of connected devices. Table 4.8 summarizes the collected results. The average set of 8.88 connected devices is mainly composed by 2.45 PCs, 2.35 Smartphones and 1.05 Tablet. Thus, the rest of the connected devices are slightly distributed among STBs (0.83), Gaming Consoles (0.53), Connected TVs (0.46), NAS (0.35), Connected Radio Stations (0.24) and Connected Audio Amplifiers (0.18). Other connected devices which are not provided in Table 4.8 represent only an average of 0.22 and are dominated by Blue Ray Players, Chromecast and Connected Printers.

4.5.3 Residential Networks Services: A Customer Point of View

Let's focus now on home network non-managed services. Our aim is to discover which applications are the most prominent while connected to a fixed Internet access. In addition, interviewed subjects are invited to depict which connected device they use for each application category. A summary of the obtained answers is presented in the following.

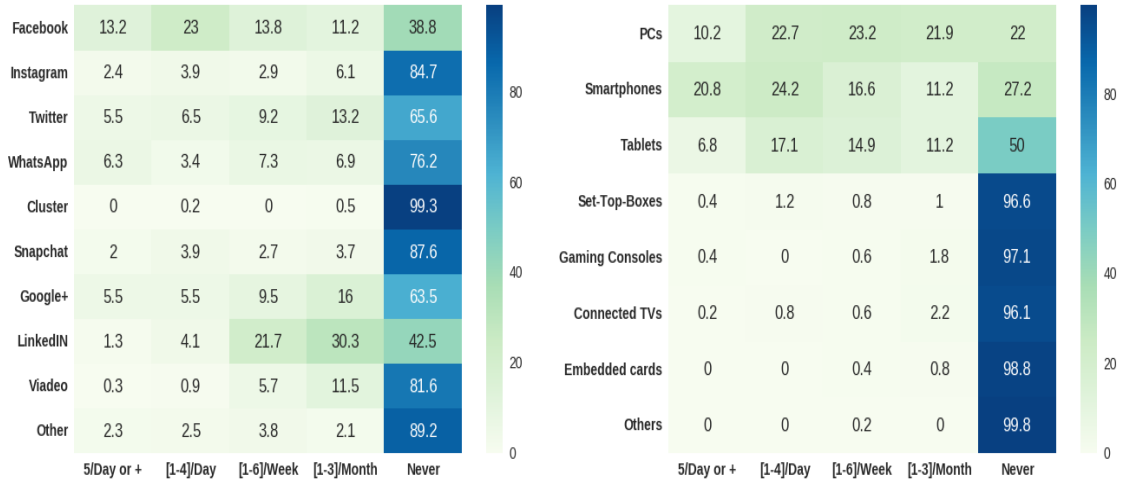

Figure 4.12 Social Networks services usages distribution (% of interviewed subjects)

4.5.3.1 Social Network services

As depicted in Figure 4.12, Facebook and LinkedIn dominates monthly usage of social network applications. In fact, 61.2% (resp. 57.5%) of subjects uses Facebook (resp. LinkedIn) at least once per month. However, usages frequency is balanced differently between both applications. While Facebook is the most used application per day (36.2% of interviewed subjects use it at least once each day), LinkedIn is used in a lighter way (weekly/monthly frequency).

Social Networks applications are more accessed through Smartphones when it comes to a daily usage. Monthly/Weekly connections to such services are coming more from PCs.

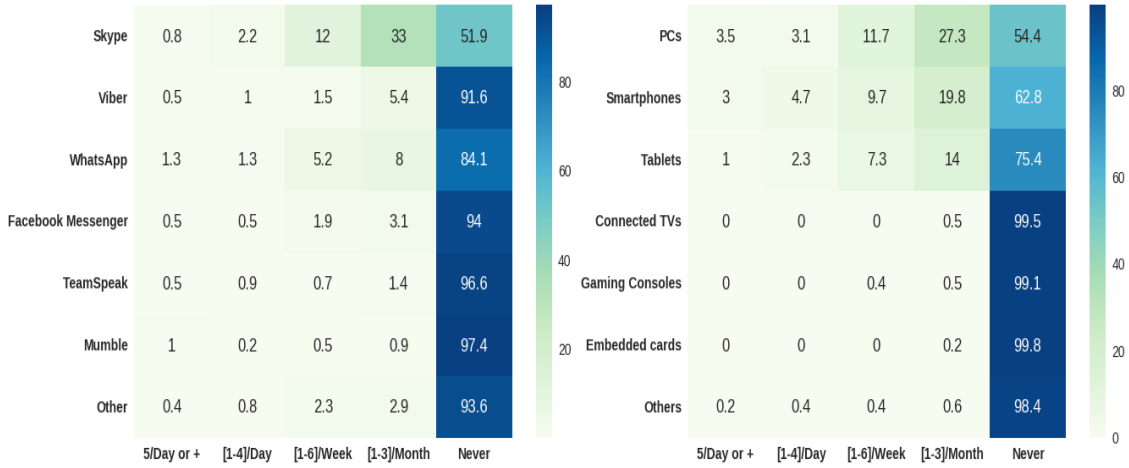

Figure 4.13 Voice Communications services usages distribution

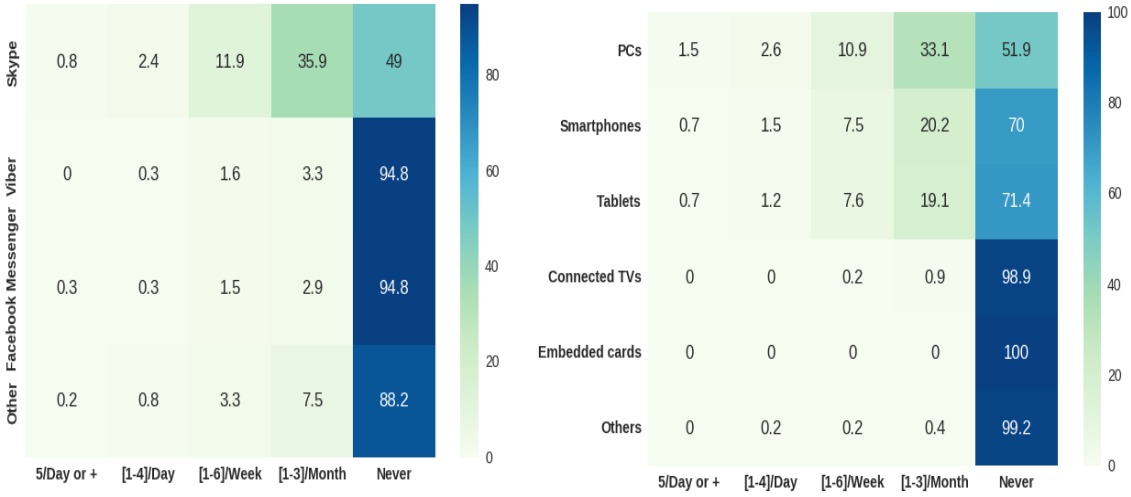

Figure 4.14 Voice Communications services usages distribution

4.5.3.2 Vocal communication services

Skype is the most used application among the voice communication services as depicted in Figure 4.13. 48.1% of interviewed subjects perform a voice call through Skype at least once

per month. PCs device usage is prevalent followed by smartphones and tablets for vocal communications.

4.5.3.3 Visio communication services

Again, Skype service is the most used one: 51% of interviewed subjects at least once per month as illustrated in Figure 4.14. Other services that are not represented in the Figure are mainly dominated by the FaceTime application. Usage among connected devices has the same trend as voice communication services with PCs being the most dominant devices. However, we observe a higher contribution of Tablets and a lower utilization of smartphones. Consequently, we assume that users tend to use tablets when performing a video communication more than smartphones for visual comfort reasons.

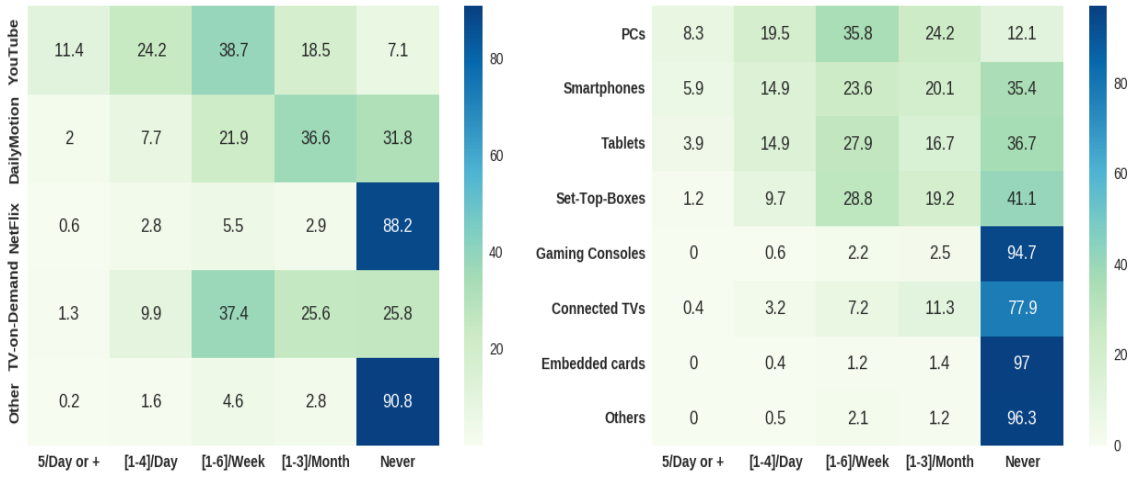

Figure 4.15 Video streaming services usages distribution

4.5.3.4 Video streaming services

Figure 4.15 illustrates the dominance of YouTube application usage. In fact, 92.9% of subjects affirm using YouTube at least once per month. TV-On-Demand platforms come in the second position followed by DailyMotion. The distribution of video streaming usages among connected residential devices unveils some new trends. While PCs, Smartphones, Tablets and Set-Top-Boxes set unsurprisingly lead the interviewed subjects' preferences, we observe that 22.1% of users start using Connected TVs at least once per month. Such trend could be explained by the recent proliferation of these advanced TV sets on the market. Furthermore, the repartition among the different devices is more balanced with respect to the previous services.

4.5.3.5 ISP Live TV services

In this section, we focus on ISP's Live TV services. Note that in addition to IPTV services, we also include in our definition, applications provided by the ISP to their customers to benefit from Live TV services (e.g. Orange TV application for smartphones and tablets). As depicted in Figure 4.16, 34.9% of interviewed subjects do not use Live TV services. Unsurprisingly, Set-Top-Boxes is prevalent as used device followed by tablets and PCs.

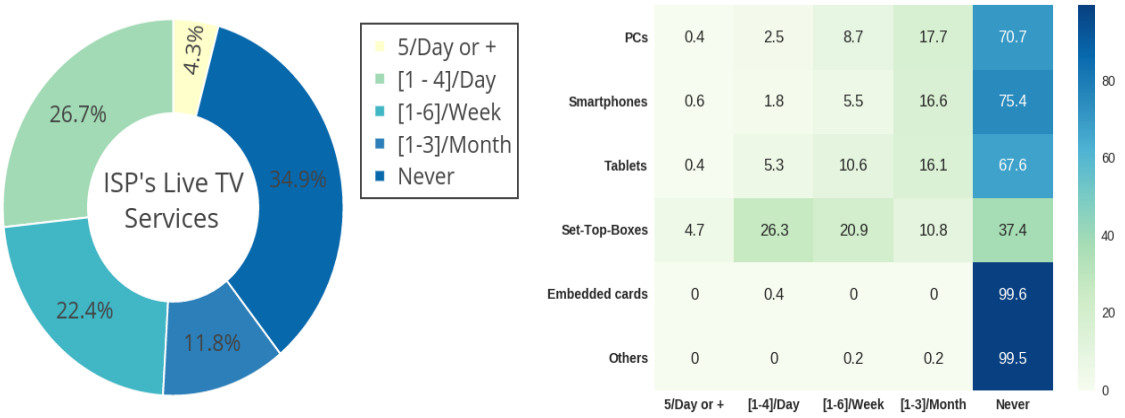

Figure 4.16 ISP's Live TV services' usages distribution

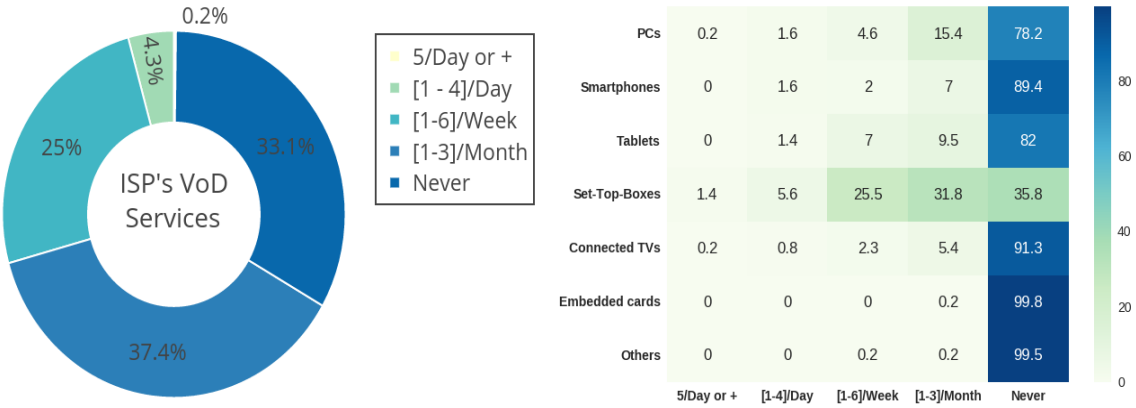

Figure 4.17 ISP's Video on Demand services' usages distribution

4.5.3.6 ISP's Video on Demand services

66.9% of subjects use Video On demand services provided by their ISP as illustrated in Figure 4.17. The observed distribution unveil that users tend to access VoD services more on a Weekly/Monthly frequency compared to Live TV services. In fact, a proportion of 62.4% of users consumes such services at a frequency range of [1-6] per week or [1-3] per month while it decreases to 34.2% for ISP's Live TV services. This trend could be explained by the interviewed population based on some French ISP employees which get some free

VoD credits. As expected, the Set-Top-Box are clearly the preferred devices for this service. We note also that PCs are preferred to Tablets unlike Live TV services case.

4.5.3.7 Audio streaming services

Deezer audio streaming platform is largely used as depicted in Figure 4.18. We observe that 45.8% of interviewed users connect at least once per month to this application while only 7% of them use Spotify at the same frequency. Other services that are not represented in the figure below are mainly composed by audio streaming services offered by radios broadcasters. The dominant trend of Deezer services is mainly explained by the interviewed population characteristics. In fact, interviewed subjects are mostly customers of the ISP that owns the Deezer platform. Consequently, the access to Deezer services is offered as part of some ISP subscribe offers. Smartphones are the most used device while connecting to such services followed by PCs and Tablets.

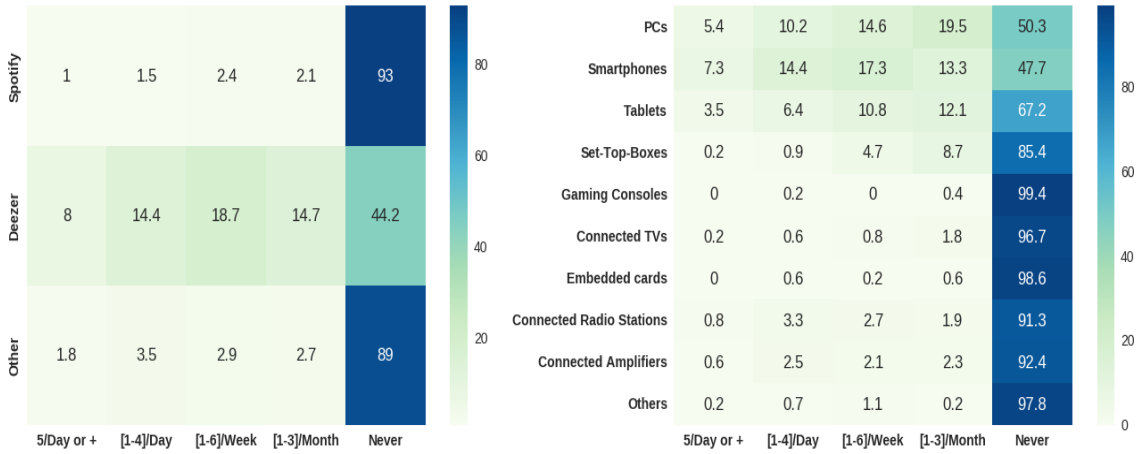

Figure 4.18 Audio streaming services' usages distribution

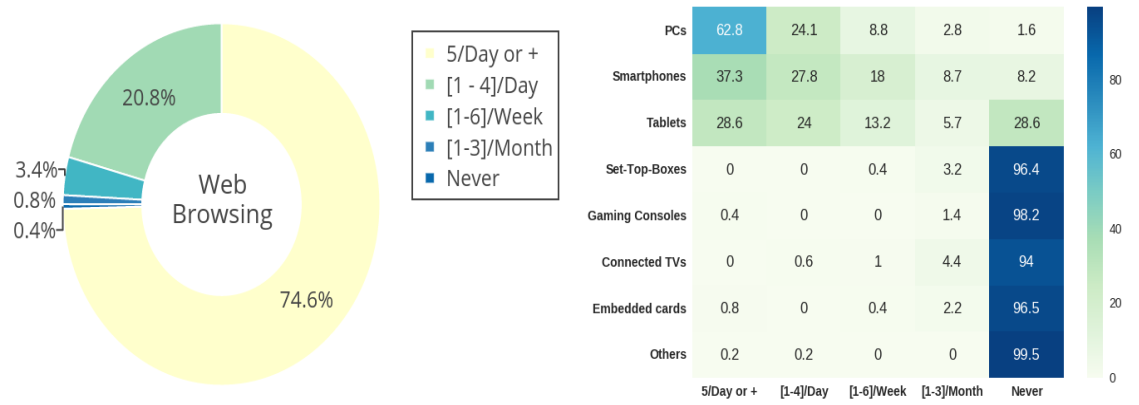

Figure 4.19 Web Browsing services' usages distribution

4.5.3.8 Web browsing services

Unsurprisingly, 95.4% of subjects navigate through Internet websites at least once per day. Subjects use mainly PCs as depicted in Figure 4.19. Smartphones are preferred to Tablets for this usage.

4.5.3.9 File Downloading services

People are asked about File Downloading habits. Answers unveil that Direct Download is used by 51.4% of interviewed population while 29.5% affirm using P2P downloading at least once per month. PCs are prevalent as the used connected device while downloading files.

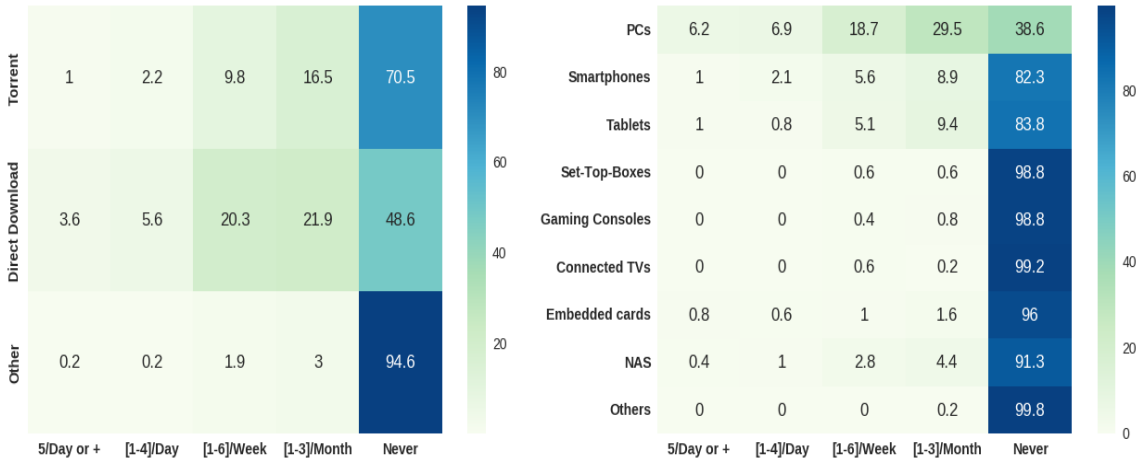

Figure 4.20 File downloading services' usages distribution

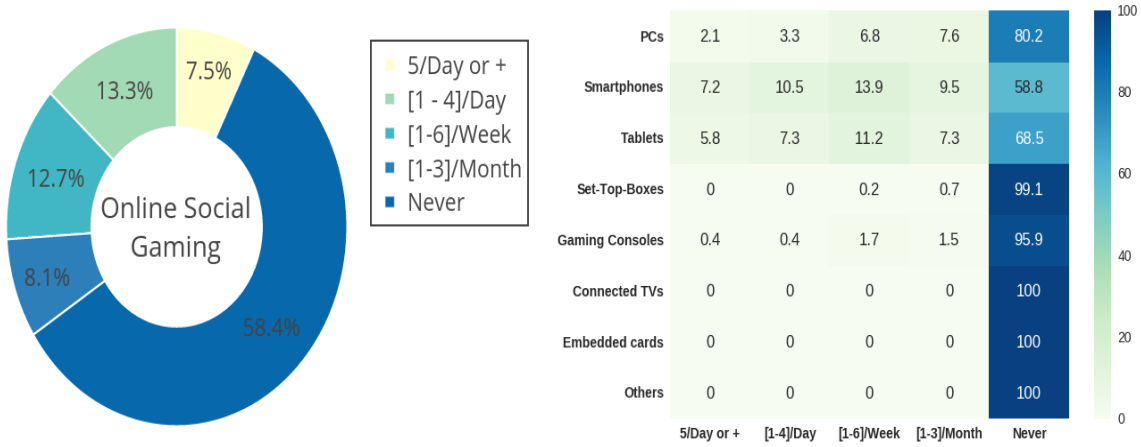

Figure 4.21 Online Social Gaming services' usages distribution

4.5.3.10 Online Social Gaming services

As we assume that the online gaming category is too coarse to tackle observed growth of social networks gaming such as CandyCrush, we decide to isolate this type of services as a separate category. 41.6% of interviewed population affirms playing on social networks at

least once per month. Such usage is mainly performed on mobile devices (Smartphones and Tablets) as illustrated in Figure 4.21.

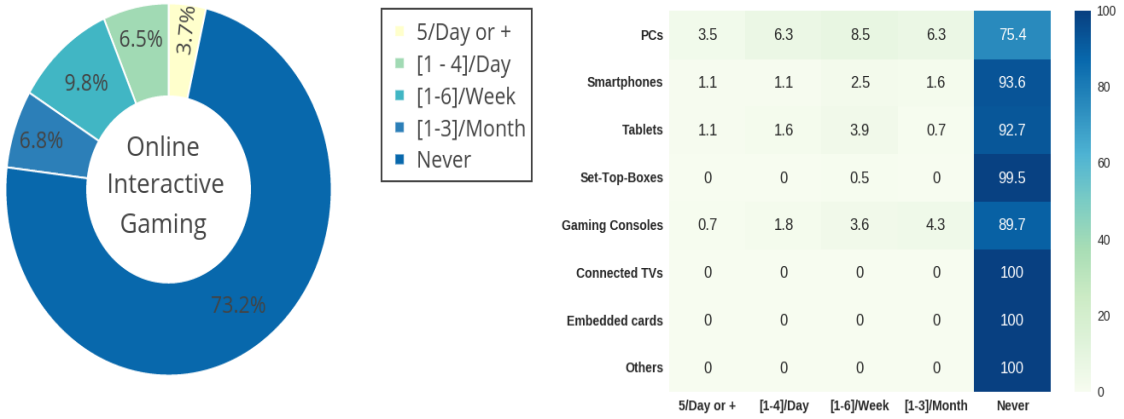

Figure 4.22 Online Interactive Gaming services' usages distribution

4.5.3.11 Online interactive gaming services

Let's focus now on online interactive gaming services. Our definition includes gaming activities that are played with multiple connected users (e.g. Call of Duty, World of Warcraft, etc.). We observe a low usage of such services. In fact, only 1 subject over 10 affirms playing online games at least once per day. As such services are usually attractive for young users (18 – 34 years old), such observation is consequently explained by the characteristics of the interviewed population (only 13.4% in this age range). Unsurprisingly, PCs and gaming consoles are the most used connected devices as illustrated in Figure 4.22.

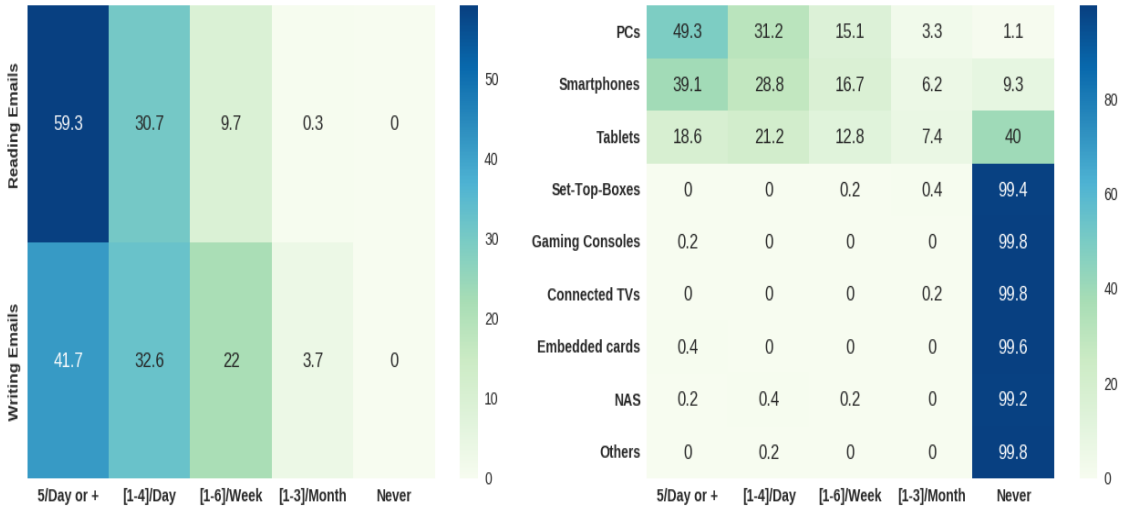

Figure 4.23 Mailing services' usages distribution

4.5.3.12 Mailing services

We focus on mailing usages of customers while connected to their fixed Internet access. Mailing services appears as the most used services among the studied application categories. While 90% of subjects affirm consulting their emails at least once per day, 84.3% affirms writing emails at the same frequency range. They use PCs as a first option and smartphones as a second one as depicted in Figure 4.23.

4.5.3.13 Online Storage services

Finally, we are interested on Online Storage services (i.e. DropBox, GoogleDrive, Orange cloud, etc.). 65.2% of studied subjects consume such services at least once per month. Their usage device source is mostly PCs. Smartphones come in the second position as shown in Figure 4.24.

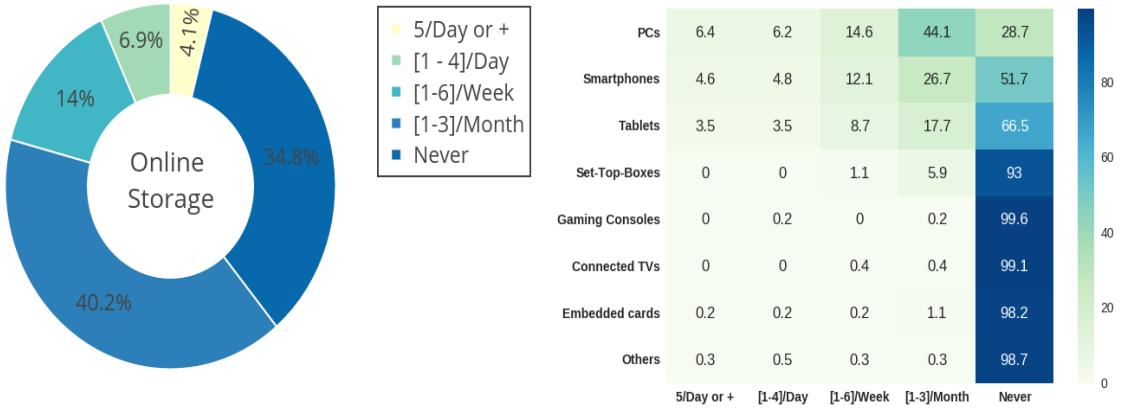

Figure 4.24 Online Storage services' usages distribution

4.6 Synthesis and Discussion

We performed a deep analysis of residential Internet usages at several scales. To the best of our knowledge, this is the first-time residential network usages are studied on both objective (Packet based) and subjective dimensions. Among our findings, 80% of the collected traffic volume is generated by 1% of the flows also known as "elephant flows". At customers' scale, heavy users' impact is observed as 5% (resp. 15%) of customers in JUL8-12 (resp. NOV26-8) trace contributing to up to 80% of the total traffic volume. This trend is more important in the uplink direction where 80% of the total volume is generated by less than 3% of the customers in the FTTH access case. This is mainly explained by Filesharing applications which represent more than 73% of such customers' uplink volume. Moreover, we found that a large proportion of Filesharing traffic is encrypted. This is mainly due to the HADOPI law

in France. In downstream, Real-time entertainment applications (e.g. YouTube, LiveTV, NetFlix) dominates the traffic in the evening while we observe a decrease at working hours period. Moreover, early deployment of OTT actor's delivery protocols such as QUIC by Google is observed. QUIC ratio increased between July and November reaching 2.7% of the overall traffic volume. Such deployment strategy may affect the known TCP/UDP ratio. Observed Google services traffic reaches 20% of the total volume at peak period (NOV26-8). Indeed, TCP represents 90% of the total traffic volume on downlink. However, on uplink, UDP generates 45% of the total traffic volume on the NOV26-8 trace and 34% for JUL8-12 trace. Finally, we observed low per-customer average link utilization.

Our reported observations are quite similar to those reported in 2013 on the same French ISP network [85] with an increasing trend regarding Filesharing applications. Comparing to Sandvine recent European reported ratios [5], we find a much higher contribution of Filesharing applications in Uplink direction. On the other hand, some applications growth such as NetFlix or Twitch is confirmed in both traces.

From a customer's standing point, our presented subjective study unveils specific characteristics of modern residential networks usages. First, home network topology complexity growth is confirmed as a typical home network tends to be composed by an average of 9 connected devices. Regarding social networks application usages, the dominance of some application such as YouTube and Facebook observed using packets analysis is confirmed. We also found that mailing services are the most daily used applications. Moreover, we leverage statistical correlation between consumed services and the corresponding device type. Customers tend to prefer the use of PCs while browsing websites or downloading files while they choose smartphone for services such as Online Social Gaming or Social Networks.

Our subjective analysis methodology is quite unique and thus, we observe a lack of works in the literature to use as landmarks. Note that Médiamétrie reported on their Home Device analysis [3] an average of 6.8 screens per household. Compared to our reported topology, results are quite similar if we exclude devices that are not plugged to a screen (Connected Radio Stations, NAS, etc.) from our 8.88 average connected devices set.

4.7 Conclusion

Exploiting a very accurate open source traffic analyzer that we enriched with our own developed tools, we presented in this chapter the results of a measurement campaign of residential

Internet traffic. The traces were collected from a major French ISP network; at a close observation point to the end users. This allows us to accurately quantify Internet customers' behavior and trends. Digging into the data at different scales, we identified common trends and patterns that allowed us to understand more in depth residential traffic. We also enriched our findings by a subjective analysis of residential customers' behavioral pattern providing a complete view of home networks usages.

Finally, despite combining several approaches and using more than 1000 application dissectors, several challenges limited our fine-grained classification process. In particular, traffic encryption and new delivery protocols are raising challenges regarding traffic identification. As we aim to provide real-time traffic classification at home network resource constrained devices (e.g. Home Gateways), we present in the next chapter our approach for overcoming classical traffic classification methods limitations. The traces capture presented in this chapter are used as an input data set for our machine learning algorithm depicted in the next chapter.

Chapter 5 Early Classification of Residential Networks Traffic using C5.0 Machine Learning Algorithm

5.1 Introduction

In the previous chapter, our large-scale analysis of a real residential traffic raised several challenges encountered using classical approaches (deep packet inspection, DNS resolving, etc.). While MLA approaches are proposed as a promising alternative, several lacks are still to be addressed as pointed out in Chapter 2. Therefore, we foresee a limited adoption of such approach in real world deployment scenario. Consequently, performing an early and reliable traffic classification is a crucial step to meet real management challenges faced by both ISP and users. Thus, value added services such as application aware QoS controller [81] or advanced parental control [82] could be efficiently designed and deployed.

In this chapter, we present an early classification approach of residential network traffic. Based on the very first packets statistical features, our approach can identify finely modern Internet services. To do so, our approach uses the C5.0 machine learning algorithm. After positioning our work in Section 5.2, we present in section 5.3 the design of our data collection and processing methodology. Then, we evaluate in section 5.4 our approach using several configurations. We address in Section 5.5 and Section 5.6 the deployment strategy of our approach. While we present the design and the implementation of our Proof of Concept

(PoC) monitoring probe in Section 5.5, we discuss our retraining process considerations in Section 5.6. Finally, we conclude in Section 5.7.

5.2 Comparative Study and Positioning

Our proposed approach aims to fulfil the seven requirements depicted in Chapter 2 namely: early input features and real-time classification, reliable dataset collection and ground-truth generation, encryption awareness, fine-grained classification output, accurate machine learning method, retraining and deployability considerations. To the best of our knowledge, this is the first time that such classification approach, combining the mentioned criteria is presented as illustrated in Table 2.3 (cf. page 35). While authors in [74] provided a post-mortem classification, our approach is real-time oriented which raises several challenges. Moreover, our dataset includes modern services (i.e. Google, Facebook, etc.) that are not considered in [74]. Furthermore, our approach is based on the C5.0 algorithm which is the evolution of the widely used C4.5. In fact, authors in [73] reported a high accuracy of C5.0 based on a synthetic dataset only. Consequently, this is the first time the C5.0 algorithm is evaluated on a real network dataset. Additionally, we developed an extension to an nProbe open source tool allowing a reliable data processing chain for features extraction. We hope that the proposed chain will be incentive to the research community to use a common benchmark. Such effort will facilitate the comparative evaluation of future contributions. Finally, our proposed approach is evaluated while deployed on a real Home Gateway hardware platform. Our results are promising in terms of resource consumption while tested using several probe/network configurations.

5.3 Data Collection and Processing Methodology

As introduced in the previous sections, the performance of statistical approaches is directly related to the used dataset.

5.3.1 Data Collection

Our data collection process is based on the two traces presented in the previous chapter. Both traces are merged in a single dataset as presented. To obtain a high-quality dataset, we perform several cleaning operations. First, we exclude TCP bidirectional flows that do not contain a SYN flag. We also exclude UDP bidirectional flows observed during the first 120s

of the capture. Such heuristics are especially useful to filter out connections initiated before the traffic capture is started and thus, ensure the extraction of the first packets statistical features correctly. The resulting dataset is presented in Table 5.1.

Table 5.1 Details of the initial dataset

Duration	2 H 33 min 49 sec		
Clients	**Total**	**xDSL (%)**	**FTTH (%)**
	34,194	15,73	84,27
Flows	**Total**	**TCP (%)**	**UDP (%)**
	41,828,024	68,13	31,25
Volume	**Total (MB)**	**Uplink (%)**	**Downlink (%)**
	371,423	19	81
Packets	**Total**	**Uplink (%)**	**Downlink (%)**
	482,188,747	41,83	58,17

5.3.2 Data Processing

Several limitations reported from the literature could be explained by the lack of a common open source chain for data processing. Despite that open source NetFlow/IPFIX probes [33, 49] exist and are sufficient to extract required features for post-mortem classification, they do not provide the early characteristics required for real-time approaches. To the best of our knowledge, only the TiE platform [83] performs such tasks. However, the ground-truth generation engine used in TiE is based on a non-reliable library (OpenDPI [66]) as identified in [68]. Moreover, being a research tool, some implemented features are considered unstable by their authors. Such observations motivate us to implement the data processing chain illustrated in Figure 5.1. In the following, we detail its major components.

5.3.2.1 Flow Features Extractor and HTTP Plugin

Flows features are obtained using the nProbe [33] (v7.3) tool allowing the extraction of the main flows' characteristics (classical 5-tuple, packets/bytes counters, etc.) in addition to some advanced performance metrics (server/network latency, etc.). We identify flows as bidirectional connections based on the classical 5-tuple {protocol ID, source IP address, destination IP address, source port, destination port}. Flow expiration is mainly driven by inactive timeout (120 sec) or natural expiration (FIN packet observed for TCP flows). Concerning the flow direction definition, we consider that data transmitted from the customer's Home

Gateway to the measurement point is counted as "uplink" while the reverse direction is called "downlink".

In addition to the common flow features obtained above, we also enabled the HTTP plugin which is available as an nProbe extension. This one performs HTTP headers extraction (i.e. URL, content type, user agent, etc.). Consequently, HTTP flows entries are enriched with their corresponding headers.

5.3.2.2 Ground-Truth Generation

The application identification engine for ground-truth generation is partly performed using the nDPI [19] library. As mentioned previously, the weakness in TiE existing platform is the reliability of one of its used libraries. Authors in [68] reported the high reliability of nDPI which outperformed the other evaluated tools. While they used nDPI version 1.6, we used in this work the following released version (1.7) which introduced several improvements such as the QUIC protocol dissector among 223 applications set. The detailed dissection logic implemented in nDPI is open source. Its main core consists of a combination of pay-load patterns matching (Aho-Corasick algorithm) and IP address mapping approach with port-based identification used as a last resort.

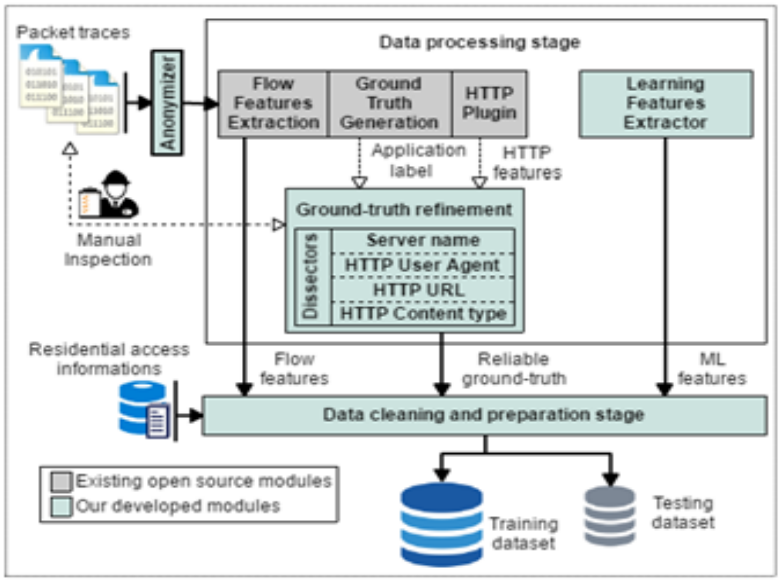

Figure 5.1 Knowledge extraction overall chain

5.3.2.3 Ground-Truth Refinement

We developed a set of parsers to refine the classification results. In fact, the used nDPI version allows defining a set of consistent application labels such as {protocol.sub-protocol} (e.g. HTTP.Facebook). We enrich sub-protocol identification for HTTP flows using

knowledge extraction from HTTP headers (HTTP User Agent, URL and content type). In addition, flows' sub-protocols identification is also affined based on resolved server name parsing for other protocols. Such approach allows us to provide in-depth view of application classes including Video on Demand, Live TV etc. which is not provided by nDPI.

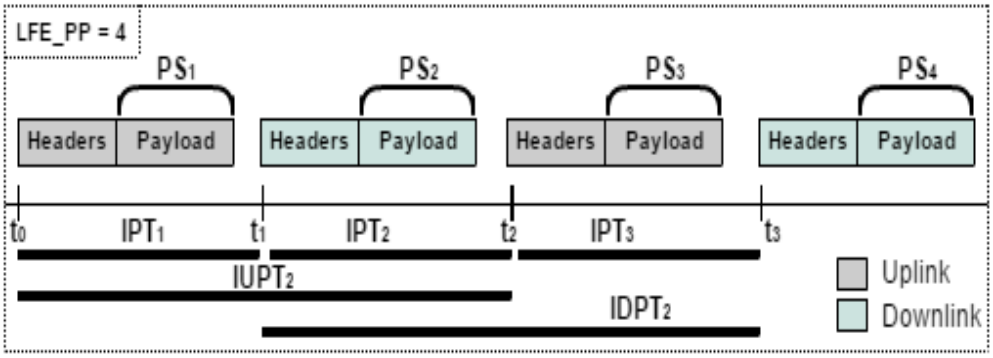

Figure 5.2 Example of LFE output per flow

5.3.2.4 Learning Features Extraction (LFE) Plugin

As we focus on early stage classification, we developed an LFE plugin which is an extension of the nProbe tool. The LFE plugin extends the extracted flow information with a set of 47 additional early learning features as depicted in Figure 5.2:

- **$PS_{1..10}$:** refers to the payload size of the i^{th} packet of the flow.
- **$D_{1..10}$:** refers to the detected direction (uplink/downlink) of the i^{th} packet of the flow.
- **$IPT_{2..10}$:** refers to Inter Packet arrival time (computed in msecs).
- **$IDPT_{2..10}$:** refers to Inter Downlink Packet arrival time (computed in msecs).
- **$IUPT_{2..10}$:** refers to Inter Uplink Packet arrival time (computed in msecs).
- **LFE_PP:** LFE plugin processed packets count.

Note that our LFE plugin implementation is designed to exclude zero payload packets and thus, observe applicative layer exchanges only. Finally, we assume that 10 packets are the maximum number of per flow observed packets to perform the early classification. Previous works showed that a threshold around 5 is usually sufficient.

5.3.2.5 Data Cleaning and Preparation Stage

Moreover, bidirectional flows for which we are not able to obtain a ground-truth label are evicted. Finally, our preparation process ensures the quality of both training and testing datasets based on three configurable thresholds:

- **Minimum-Packets-Threshold:** It sets per flow minimum observed packets threshold. For example, setting this parameter value to 6 will filter all flows having LFE_PP value < 6.
- **Minority-Class-Threshold:** Sets the minimum occurrences per traffic class to be considered. Such parameter allows us to filter minority classes as it is a known validation issue [32].
- **Cutting-Threshold:** sets the percentage of the input data to consider as training dataset. Our definition of this parameter is per-class oriented to avoid validation bias. For example, setting this parameter value to '80' will ensure that 80% of the occurrence of each predefined class will be included in the training dataset while the remaining 20% will be considered as test set. Such logic avoids the unbalanced data validation issue [32].

5.3.2.6 Summary

Combining nProbe (or forked project [84]) with our provided components results in a reliable fine-grained data processing chain. Moreover, it inherits high processing performances from the used open source tool. Finally, our output datasets (training and testing) are saved under csv format which allows to directly use it with open source ML frameworks (e.g. scikit-learn [85], WeKA [86]).

After obtaining our dataset, we propose to use the C5.0 MLA to perform early traffic classification. The conducted performance evaluation is presented in the next section.

5.4 C5.0 Classifier Performance Evaluation

5.4.1 C5.0 at a glance

The C5.0 is a new generation of decision trees based Machine Learning Algorithms. It means that the decision trees are built from applicable features extracted from the training dataset. The built tree is used to classify unknown cases (usually called testing phase). While C4.5 is identified in the literature as providing the highest accuracy, C5.0 is developed as an improved version of its ancestor [80]. Thus, the generated rules are more accurate and obtained faster (even around 360 times faster on some data sets). A detailed description of C5.0 and all its options is published in [87]. In the following, we summarize the main new techniques introduced by C5.0 editor:

- Winnowing: C5.0 winnowing routine consists of constructing a tree from the half of the data. First, the algorithm removes features that are never used as splits. Then, features that increase error rate while testing on the remaining data are filtered. Finally, it checks that the new error cost does not increase.
- Boosting: generate several classifiers (either decision trees or rulesets) rather than just one. When a new case is to be classified, each classifier votes for its predicted class and the votes are counted to determine the final class.
- New attributes: dates, times, timestamps, ordered discrete attributes.
- Missing data declaration: values can be marked as missing or not applicable for particular cases.
- Sampling and cross-validation: one-fold of cross-validation involves partitioning a sample of data into complementary subsets, performing the training on one subset, and validating the performances on the other subset (validation/testing set). Multiple folds of cross-validation are performed using different partitions, and the validation results are averaged over the folds.

The C5.0 classifier implementation [87] is open source and based on the C language. It consists of a simple command-line interface that we chained to our developed data preparation process.

5.4.2 How many packets do we need to identify a bidirectional flow?

In this section, we evaluate the C5.0 performance to define the optimum number of packets needed to achieve a reliable classification of a flow. Consequently, the evaluation dataset is prepared as follows: Cutting-Threshold = 80%, Minority-Class-Threshold = 5000 [10] flows, Minimum-Packets-Threshold = 10 packets.

Two resulting disjoint datasets are obtained to evaluate the C5.0 classifier performances. While the training dataset size is 129,973 flows, a set of 32,495 flows is used as testing dataset as illustrated in Table 3 (available packets=10). The used statistical features are $PS_{1..i}$, $D1_{..i}$, $IPT2_{..i}$, $IUPT_{2..i}$, $IDPT_{2..i}$ and the corresponding transport protocol (TCP/UDP) number. Furthermore, the classes included in our evaluation are: Facebook, BitTorrent, Skype, Google Services, QUIC, Web-Browsing (HTTP) and Secure-Web-Browsing (HTTPS). Note that for the latter classes (e.g. Web-Browsing and Secure-Web-Browsing), our definition is

[10] Computed as follow: sum(filtered_flows) / n
Where n is the number of classes existing in the filtered dataset.

fine grained as explained in Section 5.3.2.4. In fact, only flows detected as visiting simple web pages (it excludes video, audio, and media web applications) fall into those classes. We depict in Table 5.2 the performance achieved by C5.0 while we vary the number of observed packets i. For example, setting i=3 will fix the input features set as the one available when observing only 3 packets. Finally, at each execution, we run C5.0 using two modes (default and boost) to determine what the best configuration for the C5.0 classifier is.

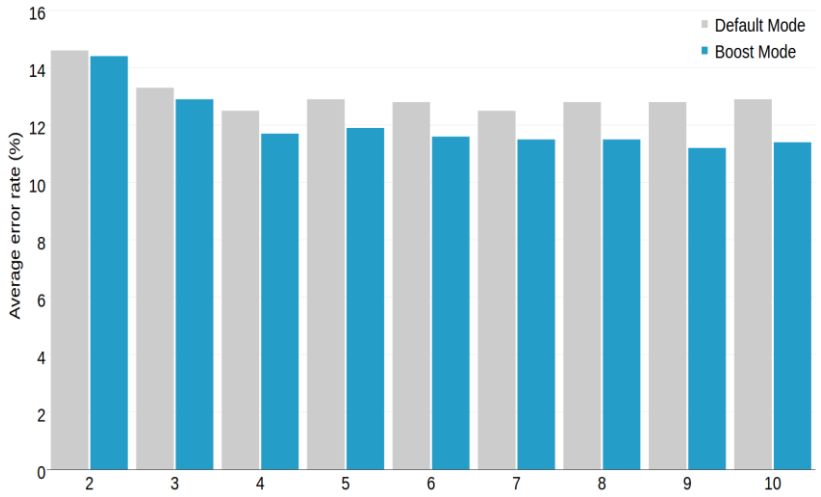

Figure 5.3 C5.0 error rate vs. min packets threshold

Table 5.2 C5.0 classification performance per observed packets threshold

		2	3	4	5	6	7	8	9	10
Training Time (secs)	**Default**	27.5	23.3	**23.8**	26.6	24.1	26	28.1	29.3	29.4
	Boost	99.9	104.3	**125.2**	146.4	162.7	189.5	210.8	240	256.1
Training accuracy (%)	**Default**	88	89.1	**89.7**	89.9	89.9	90.1	90.2	90.2	90.4
	Boost	88.4	89.8	**90.8**	91.4	91.8	92.3	92.7	93	93.4
Testing accuracy (%)	**Default**	85.4	86.7	**87.5**	87.1	87.2	87.5	87.2	87.2	87.1
	Boost	85.6	87.1	**88.3**	88.1	88.4	88.5	88.5	88.8	88.6
Generated tree size (KB)	**Default**	309	253.5	**262**	256.8	253.8	264.5	265.1	261	263.1
	Boost	1708.8	1572.7	**1850**	2012.9	2218	2541.9	2728.8	2919.3	3165.5

While we provide the accuracy measured on both training and testing datasets, we focus on the results obtained on the testing dataset. Our first observation is that the obtained accuracy is quite stable when the number of used packet features is greater than or equal to 4 on both C5.0 modes as illustrated in Figure 5.3. Furthermore, this threshold has an impact on the cost in terms of training time and generated model (tree) size as showed in Table 5.2. Thus, we conclude that 4 packets are a good trade-off in terms of accuracy vs. cost. In fact, such

threshold will allow the extraction process to check the first 4 packets only which ensures our lightweight logic in a deployment context.

Our second observation is regarding the boosted mode of C5.0. Despite a slight improvement of achieved accuracy, it dramatically increases the generated tree size (average factor of 9 compared to the default mode). Moreover, it increases also the required training time up to 9 times for 10 packets case. Such consequences, especially the size of the tree, could be a limitation in a deployment scenario where early classification is typically running on resource constrained devices (e.g. Home Gateway) which are limited in terms of memory size. Therefore, we conclude that the boost mode is not suitable in our context.

```
Attribute usage:
100%  protocol_id   98%  ipt_4     63%  iupt_2   56%  d_1
100%  ps_1          89%  dst_port  62%  iupt_4   42%  idpt_2
100%  ps_3          87%  ipt_2     58%  d_4      41%  d_2
100%  ps_2          80%  ipt_3     58%  d_3      35%  idpt_3
 99%  ps_4          63%  idpt_4    56%  iupt_3
```

Figure 5.4 C5.0 features usages (4 first packets and destination port scenario)

5.4.3 Is port number still relevant?

Despite port-based approach is criticized for its well-known unreliability; it can be relevant for certain type of traffic. Therefore, we propose to test the destination port number of a flow as an additional input feature. Consequently, the evaluation dataset is preparedusing the previously choosen values for: Cutting-Threshold (80%) and Minority-Class-Threshold (5000 flows) but while Minimum-Packets-Threshold to 4 packets. Two resulting disjoint datasets are used to evaluate the C5.0 classifier performance where the training dataset size is 266775 flows and the testing dataset size is 66698 flows as shown in Table 5.3 (available packets=4). The included classes on both datasets are the same as in the previous step.

Table 5.3 Details of the processed dataset (Minority-Class-Threshold=5000, Cutting-Threshold=80)

Available Packets	Training cases	Testing cases
4	266,755	66,698
5	218,632	54,661
6	188,708	47,181
7	167,850	41,965
8	152,452	38,117
9	139,999	35,003
10	129,973	32,495

Table 5.4 C5.0 per class performance metrics on testing dataset

	Accuracy	Precision	Recall	Accuracy: (TP+TN) / (P+N)	Precision = TP / (TP+FP)	Recall =TP / (TP+FN)
BitTorrent	99.3	98.4	98.2			
Facebook	99.3	93.4	92			
Google-services	98.4	93.1	92.9			
Web-Browsing	99.4	98.7	99.1			
Secure-Web-Browsing	97.8	88.9	93.3			
QUIC	99.5	98.3	99.6			
Skype	98.5	93.2	75.5			
Average performances	**98.8**	**94.8**	**92.9**			

- True Positive (TP): eq. with hit.
- True Negative (TN): eq. with correct rejection.
- False Positive (FP): eq. with false alarm.
- False Negative (FN): eq. with miss.
- Condition Positive (P): the number of real positive cases in the data.
- Condition Negative (N): the number of real negative cases in the data.

	Accuracy	*Precision*	*Recall*
Naive Bayes	53.3	42.6	40.5
K-NN	76.2	64.1	59.8
C4.5	98.1	94.5	92.8
C5.0 (K=1)	98.9	94.7	92.6
C5.0 (K=10)	99.0	95.3	93.4
C5.0 (K=20)	99.1	95.9	93.8
C5.0 (K=25)	99.1	95.9	93.9

Detailed performance evaluation of the C5.0 classifier applied to our input features set are depicted in Figure 5.5. We present the detailed results using the obtained confusion matrix (lines: ground-truth, columns: predictions) on the testing dataset. We also provide per class recall and precision metrics in addition to accuracy in Table 5.4.

The overall measured accuracy on the testing dataset is 98.8%. Such good performance is explained by the high-quality dataset used. Our approach can identify accurately encrypted services behind SSL such as Facebook and Google Services (99.3% and 98.4% respectively). Moreover, we identify Bittorrent flows (which is known to be a challenge for classical approaches) with over 99.3% of accuracy. Finally, despite performing high accuracy on Skype flows, the computed recall is quite low. This is mainly due to Skype flows that are classified as HTTPS. Such observation is explained by the initiation phase of Skype applications. In

fact, such phase involves HTTPS flows contacting the Microsoft platform. Those flows are classified as Skype using our ground truth generator while the extracted knowledge from HTTPS class of our classifier detects it as HTTPS.

It is worth mentioning that including destination port number as an input feature increases drastically the generated model size. In fact, port number is used as a split node in early tree building phase as showed in Figure 5.4 and the large covered set of discrete ports values implies higher number of branches. In spite of that, our results motivates using the destination port in a non-boosted configuration as it is relevant for improving our approach performances.

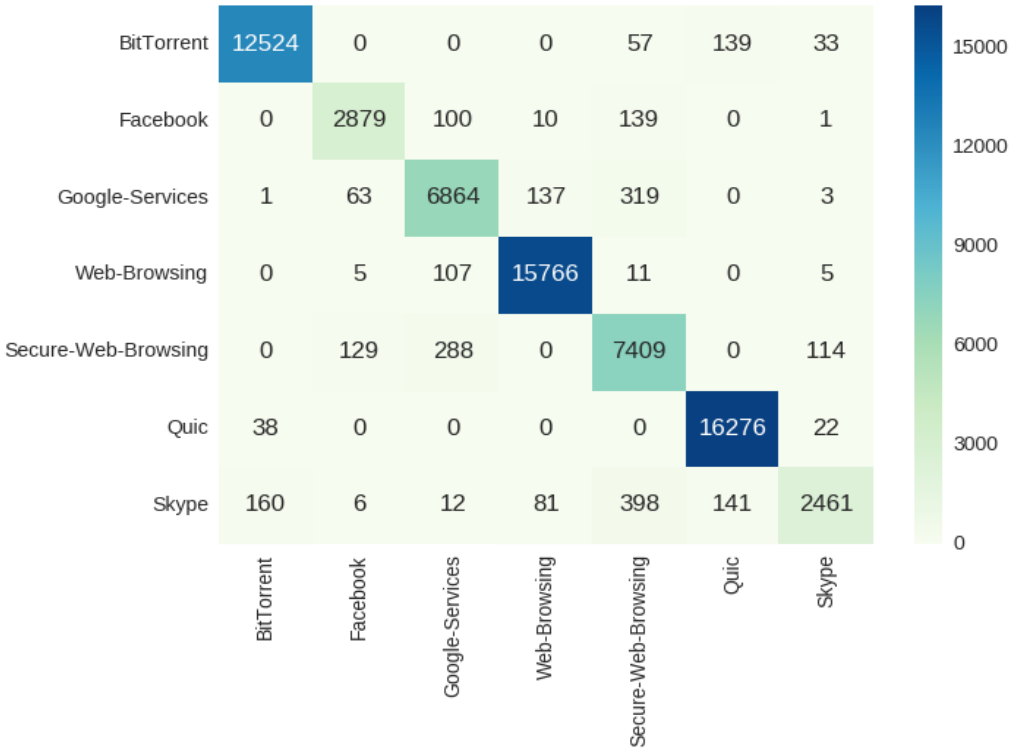

Figure 5.5 Confusion matrix of resulting classification

To the best of our knowledge, it is the first time that a detailed evaluation of C5.0 algorithm used for traffic classification is provided. In fact, authors in [74] reported high accuracy (99%) of C5.0 algorithm evaluated on a synthetic dataset.

While we provide an evaluation based on a real dataset, we also focus on early classification which is not addressed in [74]. Authors in [73] report a high accuracy while evaluating C4.5 in early classification task. However, the defined granularity is coarse and the results cannot be directly compared to ours using C5.0. Finally, these other works report performance

based on accuracy only. In our work, we present detailed metrics of our classification including recall and precision parameters. Such methodology can reveal per class performance limitations that are not reported by accuracy only (e.g. Skype case).

5.5 Experimental study

In this section, we address the deployment issue of our approach through an experimental PoC implementation of a home network traffic monitoring platform. We implemented a software probe on a home gateway prototype (having the same chipset and hardware characteristics as a commercially deployed one). The objective of this software probe is to classify at early time residential active flows using our C5.0 developed machine learning approach. The probe performs also real-time traffic monitoring and exports flow statistics.

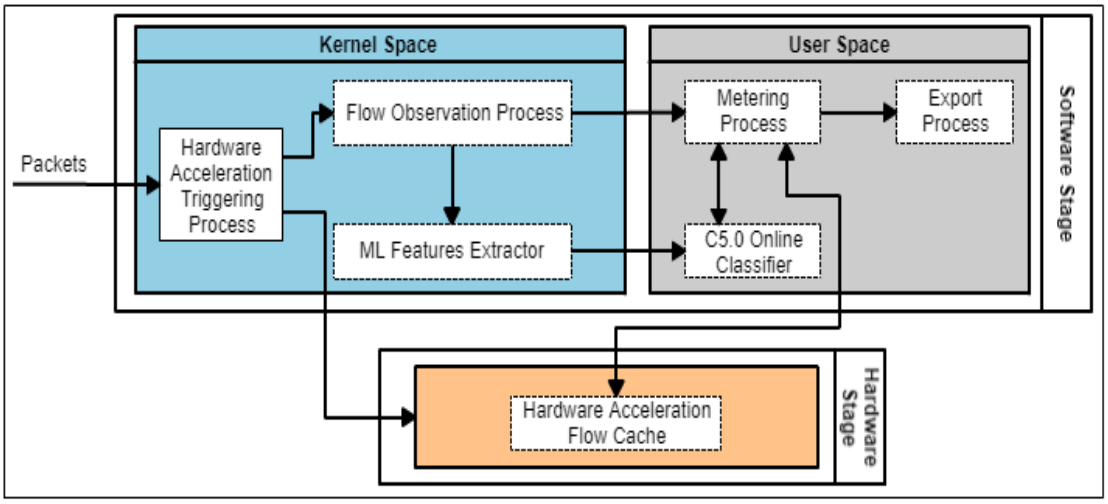

Figure 5.6 Overall design of the implemented probe

5.5.1 Overall Design

Our probe [88] is designed to fulfill the four functional requirements defined in Chapter 2 (full visibility, real-time flow monitoring, early and reliable application identification, low computational and hardware complexity). At this aim, several software modules are implemented as depicted in Figure 5.6.

- Hardware Acceleration Triggering Process: performs per flow hardware acceleration triggering. This module is responsible of delaying the hardware acceleration process by 'n' packets for each flow. While a packet index is lower than 'n', a trigger signal is sent to hardware accelerators to ignore this packet. Packets with index greater than

'n' are processed by the hardware accelerators. Consequently, released 'n' packets are observable at software level allowing to compute the necessary operations to detect the corresponding application. Indeed, hardware accelerators are used on our prototype home gateway preventing the packet observation at Linux kernel to guarantee higher routing performance.

- Flow Observation Process: Running at kernel level, this process captures and treats (extract TCP flags for example) each observable packet belonging to a given flow. Resulting output of such process is a bidirectional flow tuple, first observed packets counter, first observed packets bytes counter.
- ML Features Extractor: It extracts the C5.0 tree required features as packet sizes, inter arrival delays, direction and so on (ps_i, ipt_i, dst_port, idpt_i, iupt_i, d_i) for each flow. This process is isolated from the Flow Observation Process to facilitate the implementation update in case of new features choice.
- C5.0 Online Classifier: It consists mainly on the generated tree file and a parsing module that can parse the generated file and return the classification results. Consequently, a flow fine-grained classification result is obtained at the n^{th} packet of a flow. The tree based classifier is computed offline after the training stage, then it is embedded to our prototype home gateway.
- Hardware Acceleration Flow Cache: a flow cache is maintained by the hardware accelerators of the prototype home gateway and exposed to the kernel level through a process information pseudo-file system. It contains the current state of accelerated flows 5-tuple (IP source and destination addresses, source and destination ports and protocol), bytes counter, packets counter). When a flow expires (natural expiration, inactive timeout), the entry related to this flow is automatically flushed.
- Metering Process: The metering process takes as input flows entries transmitted by the Flow Observation Process and maintain a flow table with periodic updates of performance metrics. Considering a configurable parameter poll_period, the metering process accesses periodically the hardware acceleration flow cache maintained table to refresh per flow counters. The periodic update is performed until a flow expiration trigger is detected. A flow is considered as expired based on inactive timeout or natural expiration in TCP FIN case. Active timeout is also implemented

and is set to the same value of export periodic value. It enables real-time monitoring of active flows on the collector side.

- Export Process: export periodically the expired flows using TCP protocol. We used json format as its provide more flexibility on the collector side [89]. Note that flows that expired naturally are exported immediately. The periodic export is configurable using the export_period parameter. In addition, our architecture for export is based on the publish/subscribe paradigm. The probe is the publisher and the export is performed only if a collector subscribes to the probe.

5.5.2 Performance evaluation

In this section, we study the performance of our probe PoC implementation running on an experimental testbed. The target is to estimate the load induced by our approach under several scenarios. Our evaluation is focusing on resources consumption (CPU usage, memory usage and bandwidth load) on the Home Gateway. Furthermore, ntopng [91] was used as a collector, it is an open source collector provided by the ntop project.

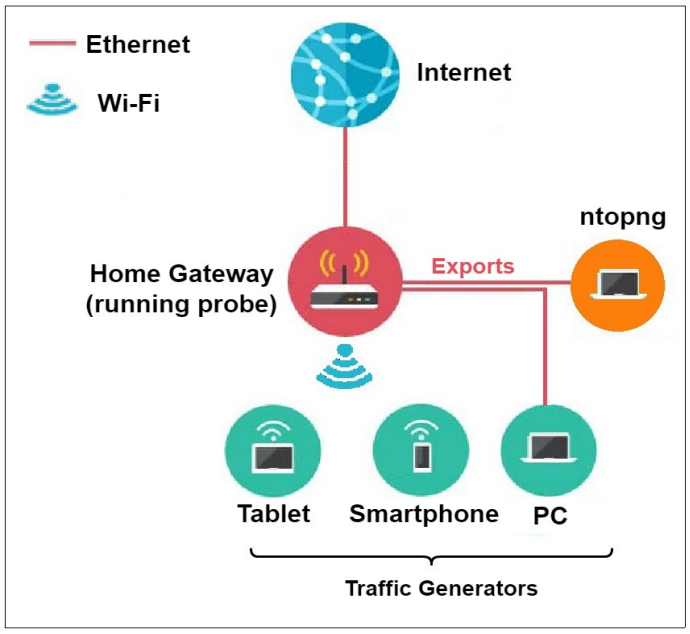

Figure 5.7 Testbed setup

5.5.2.1 Testbed setup

Our testbed is designed to fit both synthetic and real traffic scenarios test cases. We emulate several home network configuration use cases (multiple devices/OS and connectivity technologies) as depicted in Figure 5.7. Our probe is integrated on a Home Gateway prototype

having the same hardware characteristics (CPU: 2* 1.2 Ghz, Memory: 1GB) as a commercially deployed one (Orange Livebox 4). For our testbed purposes, ntopng collector runs on a local computer in the LAN. We acknowledge that a remote placement of the collector could be studied in a future work. Finally, per process resource consumption (CPU and memory) computation is performed on the Home Gateway side. Network load is measured on the collector side using the atop/netatop tool. We used both synthetic and real traffic scenarios as follows.

- Synthetic traffic scenario: The PC is used as an iperf client connected to a remote iperf server (public one). Then, we generate, as a first step, a single UDP flow (packet size of 1500 bytes) at several rates (50 Mbps, 200Mbps, 400Mbps, and 800Mbps). A second series of tests consist of setting the rate at 800Mbps while varying the number of parallel flows (1 flow at 800Mbps, 10 parallels flows at 80Mbps, 50 parallels flows at 16Mbps and 100 parallel flows at 8Mbps). Our test is limited to 100 flows due to the used public server limitation. The aim of the first series of tests is to evaluate CPU and memory usage under different flow rates while fixing the probe parameters as poll_period is set to 1 seconds and export_period is set to 2 seconds. Then, we evaluate the impact of flows number in the second series of tests. Finally, in the third series of tests, we focus on our probe parameters impact (both poll_period and export period) while setting the network traffic parameters to 100 flows at 8Mbps each. At this aim, we first fix the poll_period value to 1 second while varying the export period to 1, 2, 5, 10 and 15 seconds. In a second step, we fixed the export_period value to 5 seconds while varying the poll_period in [1, 5 seconds] range. Note that all the above test series are conducted during a 5 minutes time slice.

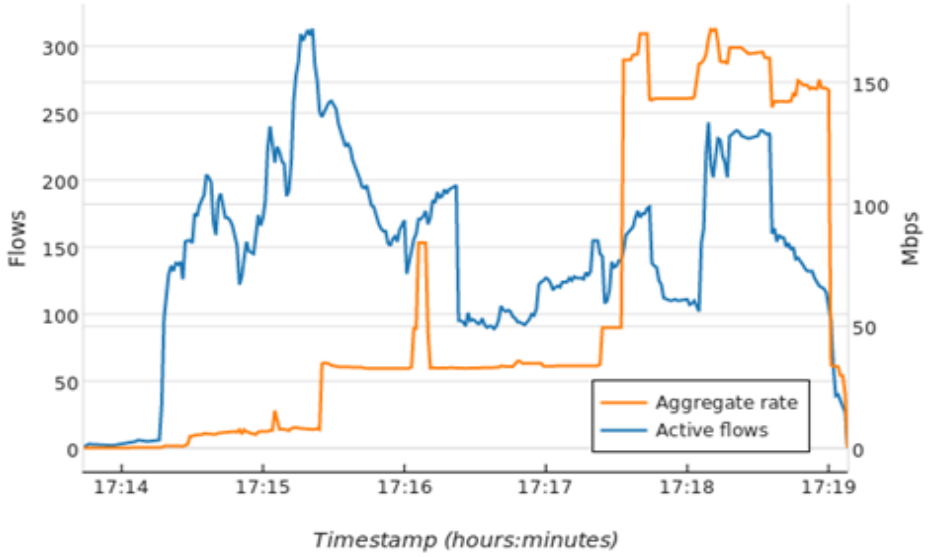

Figure 5.8 Real traffic scenario activity statistics

- Real traffic scenario: traffic generator devices (smartphone, Tablet and PC) are used to simulate a user's typical heavy load scenario (Bittorrent downloading, web browsing, playing multiple HD videos, Skype, large file downloading) with 58.73 Mbps cumulated average traffic throughput (5950 unique flows with an average of 156 flows/seconds). Generated traffic distribution is depicted in Figure 5.8. Note that the probe is evaluated using a single parameters configuration (poll_period = 2 seconds, export_period = 5 seconds) while connected to a 1Gbps FTTH access line. Test duration is kept on 300 seconds.

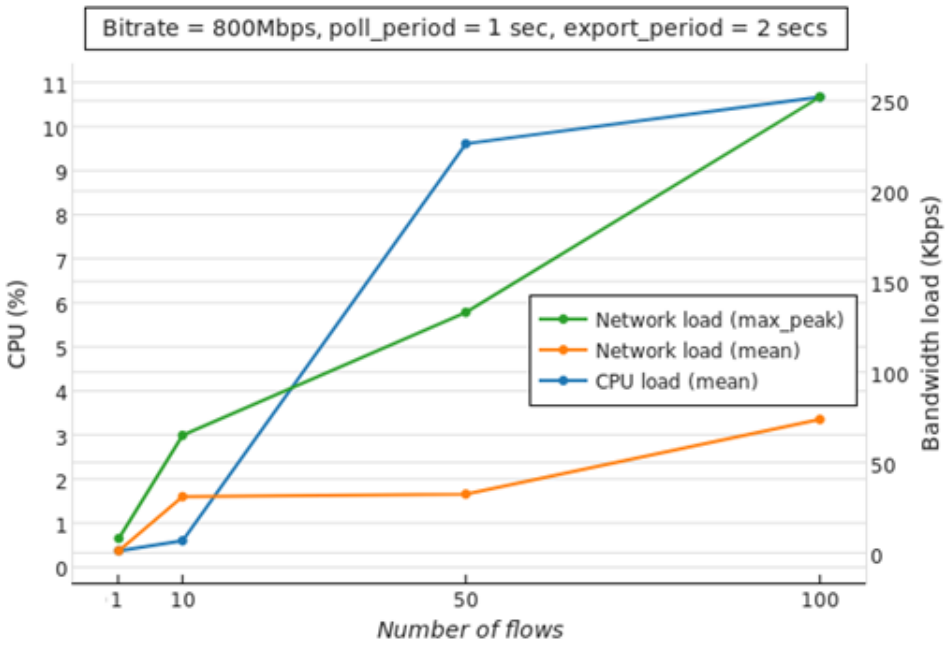

Figure 5.9 Impact of the number of flows on the probe resources consumption

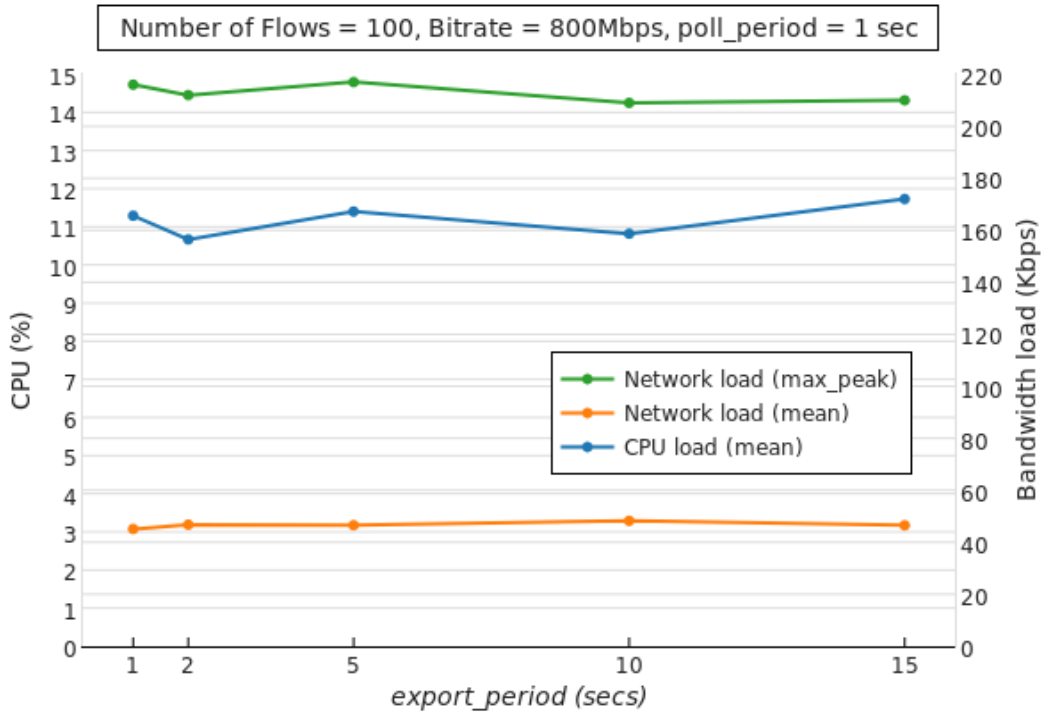

Figure 5.10 Impact of the export_period parameter on the probe resources consumption

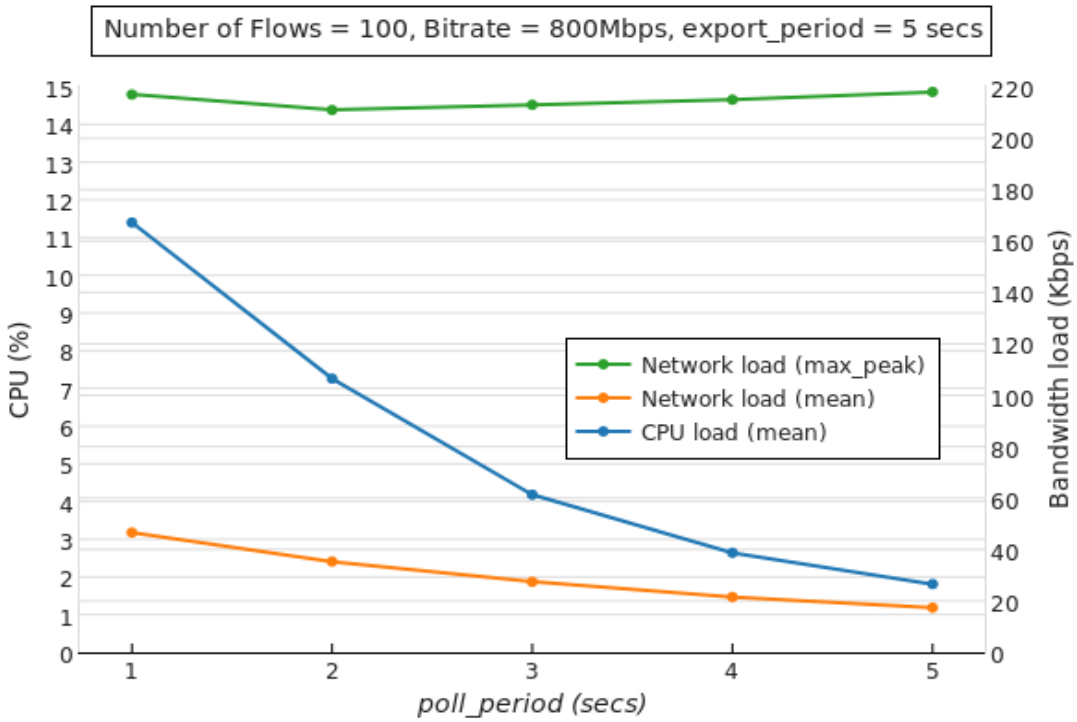

Figure 5.11 Impact of the poll_period parameter on the probe resources consumption

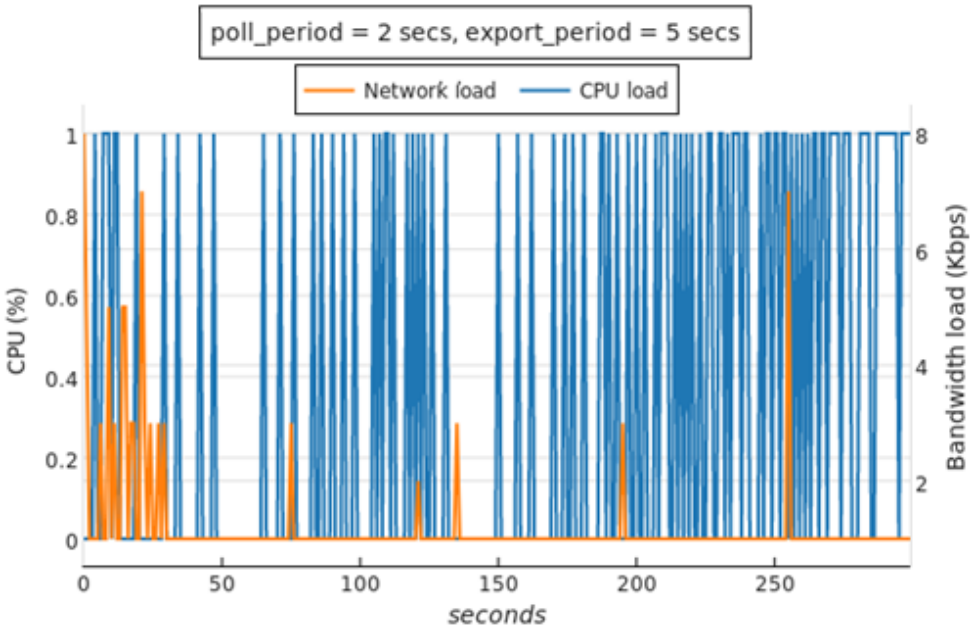

Figure 5.12 Performances evaluation using real traffic scenario

5.5.2.2 Discussion of the results

As stated above, we focus on the evaluation of our probe resource consumption based on:

- CPU usage: We measured CPU consumption using 1 UDP flow at several bitrates (50, 200, 400 and 800 Mbps). We observe that a very low consumption (maximum measured value of 1% and mean value is 0.37%) is induced. Such observation was expected due to our implementation design. In fact, only 4 packets are delayed and transmitted to the software level while remaining packets are processed at hardware level. Thus, we conclude that the bitrate does not impact our probe CPU consumption and we focus rather on the number of flows. Indeed, in a second step, we fixed the bitrate at 800 Mbps and we varied the number of parallel flows. Obtained results are depicted in Figure 5.9. While CPU consumption remains low for 1 and 10 flows cases, we observe a mean consumption of 9.61% and 10.67% for 50 and 100 flows, respectively. This is mainly explained by the metering and exporting process which is linearly solicited when increasing the number of flows. Finally, we checked the impact of the probe parameters. Under 100 flows generating an overall bitrate of 800 Mbps, we varied the export_period parameter while fixing the poll_period to 1 sec. We observe a stable consumption which shows no real impact on CPU consumption (mean consumption ~ 11%) as illustrated in Figure 5.10. In a second step, we fix the export_period to 5 seconds and vary the poll_period while keeping the same network load configuration. We observe a decreasing CPU usage trend while incrementing

the poll_period passing from a measured mean of 11.3% for 1 sec scenario to a measured mean of 1.8% for 5 secs case, as showed in Figure 5.11. Such trend is due to the metering process refresh frequency. Unsurprisingly, reducing poling activity induces lower CPU load at the expense of lower reactivity. Based on the above observation, we set our probe with poll_period = 2 secs and export_period = 5 secs which appears as a good tradeoff between monitoring accuracy (driven by poll_period), real-time observation (driven by export_period) and resource consumption. Then, we test our probe using real traffic scenario detailed in the previous section. Our results are depicted in Figure 5.12 and shows very promising performances in a real-life scenario. In fact, our probe process, classify and export a heavy user traffic load scenario flows while keeping average CPU usage under 1%.

- Memory usage: Measured memory consumption is stable and is equal to 4% in all realized tests. Memory used space is mainly consumed by the C5.0 generated model structure (tree based in our case, trained with 4 packets threshold and including port number).
- Network load: Our last evaluation focuses on the bandwidth load generated by exported data. In synthetic traffic scenario, we analyze the impact of varying the number of synthetic flows (Figure 5.9). Our evaluation confirms the trend observed for CPU usage and shows an increasing trend of network load while increasing generated flows number. For 100 flows scenario, we observe peaks up to 243Kbps where the mean measured value is 73.8 Kbps. Such observation is logical as the number of exported records is directly related to the number of monitored flows. However, we keep in mind that for FTTH access links such peaks value would have little impact on user upstream bandwidth. Finally, export parameter has a slight impact on the network load induced by the probe. Therefore, poll_period reversely impact the measured mean of network load (passing from 40 Kbps for 1 second case to 20 Kbps for 5 seconds case). This is directly related to the fact that enlarging the poll_period decreases the number of flow expiration detections.

  In real traffic scenario, we observe an average load of 1.03 Kbps with spikes up to 8Kbps corresponding to export instants as depicted in Figure 5.12. Note that our probe is configured to export only flows that have a packet counter of at least 4 packets which is the configured classification threshold.

To sum up, we can say that the probe performance, under several scenarios, are satisfactory. Indeed, L7 monitoring using heavy load real traffic needs less than 1% of average CPU usage. Moreover, memory average load is stable at 40MB. Finally, records exporting induces an average overhead of 1.03Kbps with the conducted scenario.

To illustrate the benefits of flow monitoring, we display the exported flow records on the ntopng Graphical User Interface (GUI), as depicted in Figure 5.13. Real time throughput and volume of each flow are indicated along with other information (IP addresses, duration, etc.). It is also possible to focus on a specific device to check all flows it generates and the corresponding applications. This kind of information might be useful for the end user and the ISP help-desk for troubleshooting purposes. As we can see, our probe is able to identify the QUIC protocol and the Facebook flows (with 99.9% and 96% confidence levels, respectively). Some False positives (e.g. 2 Facebook flows detected as Google with high confidence level) are also observed. We note that used classifier (i.e. C5.0 decision tree) was generated in 2015 while classified traffic is generated in 2017. Such observation can be addressed by retraining architecture as described in the next section.

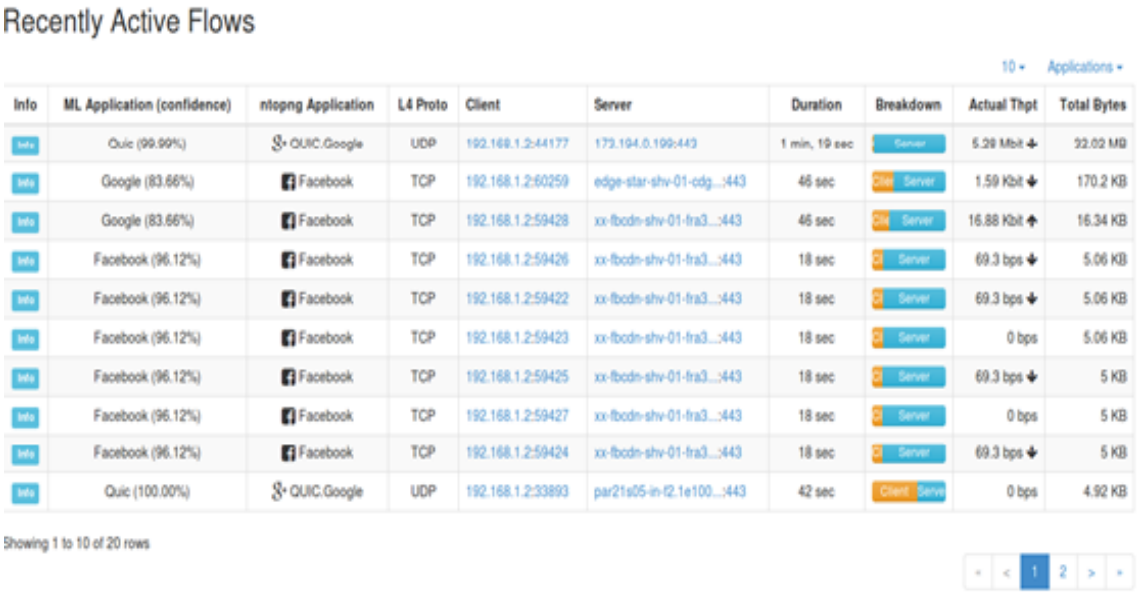

Figure 5.13 Screenshot of collector GUI, ongoing flows

5.6 Discussion: Retraining Process

In this section, we describe our retraining process proposed architecture as depicted in Figure 5.14. We believe that it is a key step to ensure the viability of an MLA on the long run (i.e. deployment context).

Based on the architecture components that we previously proposed in Chapter 3, our retraining process steps are as follows:

- The initial ML traffic classification model is deployed on the Home Gateway to ensure early classification inside the home network. As it is a resource constrained device, our approach based on the observation of the first 4 packets only as a statistical features is lightweight and could be easily achieved without performance degradation as demonstrated in the previous section.
- Once, a flow is identified after its 4th packet, the used input features, the flow 5-tuple, and the classification result are exported using IPFIX to the Home Networks Monitoring Center (cloud based).
- Ground-truth is obtained from some end-host based probes and is exported using IPFIX labelled records used as Oracle for our supervised approach. Such end-host tools ([88] for mobile platform and [63] for desktops) already exist and provide the most reliable base-truth. In fact, we assume that the retraining process should not be based on DPI or port-based approach. In fact, such choice implies the inheritance of the limitations introduced by classical approaches and thus prevents ML approach from being a real replacement of existing approaches. The deployment of end-host based ground truth generator would be performed only on a small subset of volunteer customers. Such population must be statistically representative to cover residential customers usages.
- At the Home Networks Monitoring Center side, the classification performance of the deployed trained models is continuously compared to labelled records. Such process is equivalent to a tensor to our approach and is responsible for prompting a retraining process when high False Positive rate is detected. The updated retrained model is automatically pushed to Home Gateways allowing an automatic system update.

  Note that these are the first basic ideas of our proposed retraining process. Clearly, a deeper study is needed to obtain and evaluate a more detailed process. This study could be part of future work.

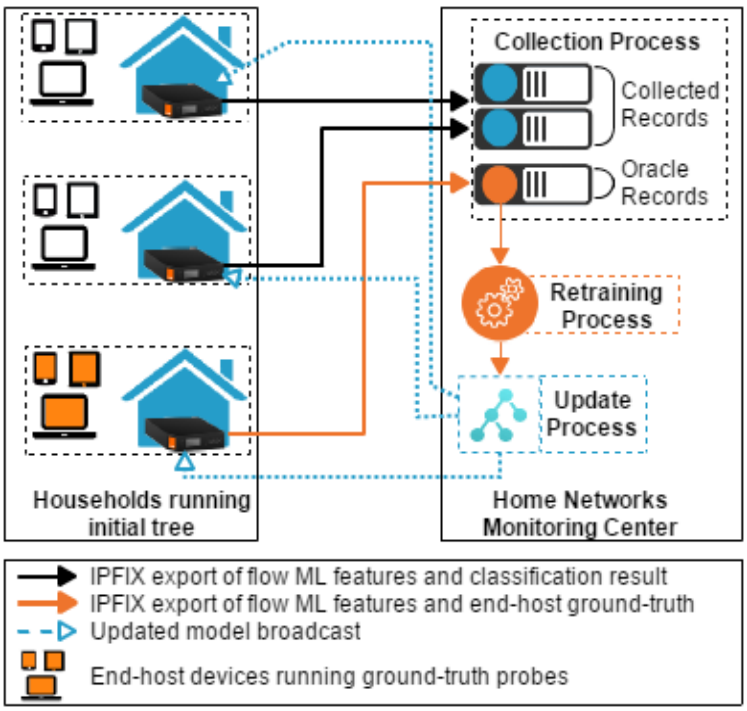

Figure 5.14 Retraining process architecture components

5.7 Conclusion

Based on the lessons learnt from the literature, we proposed in this chapter a fine-grained early classification method for residential traffic. Our approach main core is based on the C5.0 machine learning algorithm trained to identify modern Internet services in a fine-grained manner. At this aim, we detailed our developed methodology to obtain a high-quality dataset. Our approach achieves an average accuracy of 98.8% while finely classifying applications flows (e.g. Facebook, Google Services, Skype, BitTorrent, Web-Browsing and Secure-Web-Browsing) using statistical features of the first 4 packets of each flow. Performance was evaluated using advanced metrics based on a disjoint testing dataset involving more than 34,000 residential customers. Consequently, we think that our results are more convincing than previously reported ones based on a synthetic single user dataset. Moreover, we provide the community with an extension which, integrated with open source components, provide a reliable data processing chain.

Furthermore, we implemented a proof of concept of a probe fulfilling the requirements of our approach on a prototype Home Gateway. Our probe was designed to deal efficiently with hardware constraints inherited from such a platform. Thus, we validated our implemen-

tation by setting an experimental testbed. Our obtained results demonstrate that our approach is feasible and efficient. In fact, the probe can monitor, classify and track statistics of active flows in the home networks while inducing a low resource overhead.

Finally, we ensure the viability of our approach by discussing how a retraining process can be set up in order to address ML deployment issue. To the best of our knowledge, this is the first time that such classification approach, combining the above-mentioned design principles, is presented.

Chapter 6 Conclusion

Home network complexity is increasing with the multiplication and diversification of devices, services and connectivity technologies (mainly Ethernet, Wi-Fi, PLC and MoCA). Moreover, various application types are generated: Facebook, YouTube, Skype, Bittorrent, etc. In this context, when service degradation occurs, it is difficult for both the end user and the ISP help-desk to easily tackle the issue. In fact, usually, the customer tends to call the ISP help-desk even if the problem is outside the home network. On the other hand, ISPs control all segments of their networks (core, access, etc.) but not the home network portion which is becoming the most fragile one. Therefore, relying on efficient traffic monitoring tools allowing to observe and to identify home network flows is a key aspect for diagnostic enhancement and network performance improvement.

In fact, home network traffic monitoring would allow among other functions:

- Having better insight on network usage (e.g. devices consuming the highest bandwidth, flows rates, etc.)
- Applying advanced parental control (e.g. blocking access to a specific application from a given device)
- Deployment of QoS mechanisms (e.g. application based prioritization)
- Anomaly detection (e.g. botnets attacks)

However, home network traffic monitoring raises many challenges. In this dissertation, we addressed those challenges, especially in terms of feasibility. Our contributions could be outlined as follows:

- First, we proposed a novel architecture for traffic monitoring in Home Networks using flow export approach based on probe and collector components. We provided a comparative study of existing open source probe tools. Then, we performed a

benchmark of the nProbe tool evaluating the computational overhead of each performed process. Based on experiment results, we highlighted several limits preventing real-life deployment. In particular, we pointed out a crucial need for a reliable, lightweight and viable traffic classification approach.

- Then, we presented a large scale residential traffic and usages analysis based on real traces collected at a major French ISP involving more than 34,000 customers. At this aim, we developed a reliable methodology for data collection and processing. Then, we analyzed customers behavior with respect to TCP/IP protocol stack layers characteristics. Moreover, we conducted a subjective analysis across 645 residential customers. Our findings provided a complete synthesis of residential usage patterns and applications characteristics.
- The extracted knowledge was used as a corner stone to build a novel traffic monitoring and classification approach based on the C5.0 MLA. We proposed an early, fine-grained traffic classification approach. Our classifier performance accuracy was evaluated and validated on a disjoint test dataset. We implemented the proposed solution on a prototype Home Gateway proving its feasibility. Finally, we discussed retraining process which will enable to ensure the temporal viability of our approach.

In a home network context, the traffic characteristics are evolving rapidly due to the constant emergence of new applications and services. Maintaining the reliability and the efficiency of traffic monitoring and classification tools, in such context, is crucial. So, we need more reliable, viable and less intrusive monitoring techniques. The ultimate goal is an autonomic management of the end-to-end quality of experience of sensitive and critical applications. Our work presented here is a first step toward this aim.

We see a promising field of work for residential traffic monitoring and classification. Future work can include different studies such as:

- Extending our test bed evaluation with a collector deployed in the cloud. The goal will be to study possible bottlenecks implied by such architecture. The network load induced by millions of Home Gateways must be considered properly in terms of collection network dimensioning and storage challenges.
- In this dissertation, our machine learning model was tested mainly using an offline collection process. A next step could be to deploy such model on several Home Gateways and to analyse its online classification performance. Such analysis will address one of the most common validation issues observed in the literature.

- Some improvements could be done in future work on both implementation and evaluation parts of our PoC implementation. In our current implementation version, our metering process expiration detection mechanism is mainly based on an active polling of hardware accelerator flow cache. Such approach has an impact on the resource consumption, especially, with a high number of flows. One possible improvement is to change our expiration detection mechanism by sniffing expiration events using the conntrack tool. Another potential improvement is related to the performance evaluation. In the synthetic scenario, we used 100 as a maximum number of flows. Such limitation was due to iperf used public server limitation. One future work could be to deploy our remote server and thus, provide an extensive evaluation with higher number of flows.
- While we proposed a monitoring and a classification approach, we need to keep in mind that such approach is a component of a larger pipeline as explained in Chapter 2. The goal is to exploit the exported records to improve the user experience and to facilitate Home Network diagnosis process. A future work could be based on Home Network connected devices traffic profiling to detect network anomalies. IoT devices security known vulnerability could be addressed as a use case. For example, combining device type (e.g. IP camera) and the traffic patterns (streaming video, etc.) could be a first step in an outlier detection system.
- Home network monitoring architecture could be enlarged with virtualization possibilities. An architecture based on a virtual Home Gateway could include a subset of pre-analyzing and aggregation components. Thus, in-line management rules could be set without exporting records to a third-party component.
- Network load (bandwidth overhead induced by the probe) possible bottleneck could be investigated more in depth. The impact of optimization techniques such as flow sampling on the quality of the exported data could provide some interesting insights.
- Our analysis provided in Chapter 4 could be extended to a larger temporal and spatial distribution. Monitoring several customers from several observation points and during a larger period will provide additional insights that may be difficult to extract on a few hours' time slice.

Furthermore, other perspectives could include:

- A community-oriented ground-truth generation platform. End-host based tools which were highlighted as the most accurate existing solution could be improved. In fact, current ground-truth generation is based on the process name label. Despite

being considered as fine-grained, such technique faces several limits when dealing with web activities. For example, a 'Mozilla Firefox' label is useful to identify the running application. However, identifying the services running from such application seems to be trickier. The goal is to provide the community with a multilevel reliable labelling method.

- Multi-classification could be investigated to build a multi-layer retraining architecture. The use of both post-mortem and early classification with different granularities for each classifier seems to be a promising approach.

Appendix A Survey of Customers' Residential Usages

A.1 Overview of the study

The study was conducted using an online form sent via Email. Subjects were invited to answer a set of questions to unveil their behavioral habits while connected through their fixed access to the Internet. In this appendix, we detail the survey process.

Table A.1 Examples of Home Network devices

Devices included in this experiment	Devices excluded in this experiment
PCs (Desktop, Laptop, Mini), smartphone, tablet, TV decoder, gaming console, smart TV, connected radio, NAS (Network Attached Storage), embedded cards (Raspberry Pi, Beagle Board, etc.), connected (IP) audio amplifier (non-Bluetooth), etc.	Smart watch, Presence sensor, smoke sensor, Bluetooth speakers, network switch, PLC plug, Home Automation Box, IP camera, Wi-Fi extender, LivePlug Wi-Fi, smart body scale, connected printer, etc.

A.1 Introduction and Overall Context

In order to improve the management process of Home Networks, a crucial step is to understand the user's usages according to the connected devices in their Home Networks. This survey aims to study the importance of the device's type (laptop, smartphone, tablet, etc.) and its impact on the daily habits. The goal of this experiment is to build a more efficient Home Network management process to improve the quality of experience.

The questionnaire is related to the Home Network usages. This implies the usages, when connected at home through the fixed access provided by the ISP. In the following, we detail, all the asked questions.

A.1 Panel Overview

Your age?

<25 ☐
25-28 ☐
28 - 35 ☐
36 - 45 ☐
46 – 55 ☐
> 55 ☐

Your household members' number ☐

Your gender

Male ☐
Female ☐

A.2 Home Network Topology

List the devices included in the experiment (detailed in the Table.B.1) that are present in your current Home Network configuration.
Example: 2 Smartphones, 1 Tablet, 2 PCs/laptops, 1 TV Decoder, 1 Gaming Console

PC (Desktop, laptop, mini PC) ☐
Smartphones ☐
Tablet ☐
TV decoder ☐
Gaming console ☐
Smart TVs ☐
Embedded cards ☐
NAS ☐
Connected Radio ☐

IP Audio Amplifier	☐
Others	☐

Total number of listed devices (including «Others») ☐

A.3 Home Network Services

A.3.1 Social Networks Services

At which frequency do you use social networks services (i.e. Facebook, Instagram, Twitter, WhatsApp, Snapchat, Google+, LinkedIn, Viadeo, Cluster, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
Facebook	☐	☐	☐	☐	☐
Instagram	☐	☐	☐	☐	☐
Twitter	☐	☐	☐	☐	☐
WhatsApp	☐	☐	☐	☐	☐
Cluster	☐	☐	☐	☐	☐
Snapchat	☐	☐	☐	☐	☐
Google+	☐	☐	☐	☐	☐
LinkedIn	☐	☐	☐	☐	☐
Viadeo	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

Which device do you use to connect to social networks (i.e. Facebook, Instagram, Twitter, WhatsApp, Snapchat, Google+, LinkedIn, Viadeo, Cluster, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Gaming Consoles	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐

Other	☐	☐	☐	☐	☐

A.3.2 Vocal Communication Services

At which frequency do you use vocal communication services (Skype, Viber, WhatsApp, Facebook Messenger (voice call), TeamSpeak, Mumble, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
Skype	☐	☐	☐	☐	☐
Viber	☐	☐	☐	☐	☐
WhatsApp	☐	☐	☐	☐	☐
Facebook Messenger	☐	☐	☐	☐	☐
TeamSpeak	☐	☐	☐	☐	☐
Mumble	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

Which device do you use to connect to vocal communication services (Skype, Viber, WhatsApp, Facebook Messenger (voice call), TeamSpeak, Mumble, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Gaming Consoles	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.3 Visio Communication Services

At which frequency do you use Visio communication services [Vocal + Video] (i.e. Skype, Viber, Facebook Messenger (video call), etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
Skype	☐	☐	☐	☐	☐
Viber	☐	☐	☐	☐	☐
Facebook Messenger	☐	☐	☐	☐	☐

Other	☐	☐	☐	☐	☐

Which device do you use to connect to Visio communication services [Vocal + Video] (i.e. Skype, Viber, Facebook Messenger (video call), etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.4 Video Streaming Services

At which frequency do you use video streaming services (i.e. YouTube, DailyMotion, NetFlix, Video on Demand (MyTF1, M6Replay, etc.), etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
YouTube	☐	☐	☐	☐	☐
NetFlix	☐	☐	☐	☐	☐
DailyMotion	☐	☐	☐	☐	☐
Video on Demand	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

Which device do you use to connect to video streaming services (i.e. YouTube, DailyMotion, NetFlix, Video on Demand (MyTF1, M6Replay, etc.), etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Gaming Consoles	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐

Other	☐	☐	☐	☐	☐

A.3.5 ISP Live TV services

At which frequency do you use your ISP Live TV services?

5/day or +	1-4/day	1-6/week	1-3/month	Never
☐	☐	☐	☐	☐

Which device do you use to connect to your ISP Live TV services?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.6 ISP Video on Demand Services

At which frequency do you use your ISP Video on Demand services?

5/day or +	1-4/day	1-6/week	1-3/month	Never
☐	☐	☐	☐	☐

Which device do you use to connect to your ISP Video on Demand services?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.7 Audio Streaming Services

At which frequency do you use audio streaming services (i.e. Spotify, Deezer, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
Spotify	☐	☐	☐	☐	☐
Deezer	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

Which device do you use to connect to audio streaming services (i.e. Spotify, Deezer, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Gaming Consoles	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Connected Radio	☐	☐	☐	☐	☐
IP Audio Amplifier	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.8 Web Browsing Services

At which frequency do you use Web Browsing services (i.e. web sites, blogs, forums, etc.)?

5/day or +	1-4/day	1-6/week	1-3/month	Never
☐	☐	☐	☐	☐

Which device do you use to connect to Web Browsing services (i.e. web sites, blogs, forums, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐

TV decoders	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.9 File Downloading Services

At which frequency do you use File Downloading services (i.e. torrents, direct download, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
Torrents (P2P, etc.)	☐	☐	☐	☐	☐
Direct download	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

Which device do you use to connect to File Downloading services (i.e. torrents, direct download, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Gaming Consoles	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
NAS	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.10 Online Social Gaming

At which frequency do you use Online Social Gaming services (i.e. Candy Crush, Angry Birds, Social Networks Gaming, etc.)?

5/day or +	1-4/day	1-6/week	1-3/month	Never
☐	☐	☐	☐	☐

Which device do you use to connect to Online Social Gaming services (i.e. Candy Crush, Angry Birds, Social Networks Gaming, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Gaming Consoles	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.11 Online Interactive Gaming Services

At which frequency do you use Online Interactive Gaming services (i.e. Call of Duty, League of Legends, World of Warcraft, etc.)?

5/day or +	1-4/day	1-6/week	1-3/month	Never
☐	☐	☐	☐	☐

Which device do you use to connect to Online Interactive Gaming services (i.e. Call of Duty, League of Legends, World of Warcraft, etc.)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Gaming Consoles	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.12 Mailing Services

At which frequency do you use Mailing services (e.g. Writing/Reading Emails)?

5/day or +	1-4/day	1-6/week	1-3/month	Never
☐	☐	☐	☐	☐

Which device do you use to connect to Mailing services (e.g. Writing/Reading Emails)?

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Gaming Consoles	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

A.3.13 Online Storage Services

At which frequency do you use Online Storage services (i.e. le Cloud d'Orange, Dropbox, Google Drive, etc.)

5/day or +	1-4/day	1-6/week	1-3/month	Never
☐	☐	☐	☐	☐

Which device do you use to connect to Online Storage services (i.e. le Cloud d'Orange, Dropbox, Google Drive, etc.)

	5/day or +	1-4/day	1-6/week	1-3/month	Never
PCs	☐	☐	☐	☐	☐
Smartphones	☐	☐	☐	☐	☐
Tablets	☐	☐	☐	☐	☐
TV decoders	☐	☐	☐	☐	☐
Gaming Consoles	☐	☐	☐	☐	☐
Smart TVs	☐	☐	☐	☐	☐
Embedded cards	☐	☐	☐	☐	☐
Other	☐	☐	☐	☐	☐

Bibliography

[1] D. E. Meddour, A. Kortebi and R. Boutaba, "Mesh-based broadband home network solution: setup and experiments," *International Conference in Communications (ICC),* pp. 1-5, May 2010.

[2] L. Roberts, "Beyond Moore's law: Internet growth trends," *Computer,* pp. 117-119, January 2000.

[3] Le Syndicat National de la Publicité Télévisée (SNPTV), "Le guide du SNPTV 2016," September 2015. [Online]. Available: https://www.snptv.org/wp-content/uploads/2015/09/GUIDE_SNPTV2016_dpq_BD1.pdf. [Accessed 5 January 2017].

[4] Verizon, "State of the Market: Internet of Things 2016," 2016. [Online]. Available: https://www.verizon.com/about/sites/default/files/state-of-the-internet-of-things-market-report-2016.pdf. [Accessed 14 February 2017].

[5] Sandvine, "2015 - Global Internet Phenomena Asia-Pacific & Europe," September 2015. [Online]. Available: https://www.sandvine.com/downloads/general/global-internet-phenomena/2015/global-internet-phenomena-report-apac-and-europe.pdf. [Accessed 29 April 2016].

[6] R. E. Grinter, W. K. Edwards, M. Chetty, E. S. Poole, J. Sung, J. Yang, A. Crabtree, P. Tolmie, T. Rodden, C. Greenhalgh and S. Benford, "The ins and outs of home networking: The case for useful and usable domestic networking.," *ACM Transactions on Computer-Human Interaction (TOCHI),* vol. 16, no. 2, pp. 8-32, June 2009.

[7] S. Kiesler, B. Zdaniuk, V. Lundmark and R. Kraut, "Troubles with the Internet: The dynamics of help at home," *Human-Computer Interaction,* vol. 15, no. 4, pp. 323-351, 2000.

[8] S. Sundaresan, Y. Grunenberger, N. Feamster, D. Papagiannaki, D. Levin and R. Teixeira, "WTF? Locating performance problems in home networks," Georgia Institute of Technology, Georgia, 2013.

[9] Autorité de Régulation des Communications Electroniques et des Postes (ARCEP), "QoS scorecards for accessing fixed line services," 13 April 2016. [Online]. Available: http://www.arcep.fr/index.php?id=10606&L=1. [Accessed 8 December 2016].

[10] A. Kortebi, P. Le Dain and F. Dure, "Home network assistant: Towards better diagnostics and increased customer satisfaction," *Global Information Infrastructure Symposium (GIIS),* pp. 1-6, October 2013.

[11] A. Kortebi, Z. Aouini, M. Juren and J. Pazdera, "Home Networks Traffic Monitoring Case Study: Anomaly Detection," *Global Information Infrastructure and Networking Symposium (GIIS),* pp. 1-6, October 2016.

[12] A. Pekár and M. Chovanec, "Survey of the issues surrounding network traffic monitoring," *Wireless Communications, Networking and Mobile Computing (WiCOM 2015),* pp. 1-8, September 2015.

[13] D. Joumblatt, R. Teixeira, J. Chandrashekar and N. Taft, "HostView: Annotating end-host performance measurements with user feedback," *ACM SIGMETRICS Performance Evaluation Review,* vol. 38, no. 3, pp. 43-48, 2011.

[14] T. Karagiannis, E. Athanasopoulos, C. Gkantsidis and P. Key, "HomeMaestro: Order from Chaos in Home Networks," Microsoft Research, Cambridge, 2008.

[15] L. DiCioccio, R. Teixeira and C. Rosenberg, "Measuring home networks with homenet profiler," *International Conference on Passive and Active Network Measurement,* pp. 176-186, March 2013.

[16] C. Kreibich, N. Weaver, B. Nechaev et V. Paxson, «Netalyzr: illuminating the edge network,» *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement,* pp. 246-259, November 2010.

[17] A. Reggani, F. Schneider and R. Teixeira, "Tracking application network performance in Home Gateways," *Wireless Communications and Mobile Computing Conference (IWCMC),* pp. 1150-1155, July 2013.

[18] S. Sundaresan, S. Burnett, N. Feamster and W. De Donato, "BISmark: A Testbed for Deploying Measurements and Applications in Broadband Access Networks," *USENIX Annual Technical Conference,* pp. 383-394, June 2014.

[19] L. Deri, M. Martinelli, T. Bujlow et A. Cardigliano, «ndpi: Open-source high-speed deep packet inspection,» *Wireless Communications and Mobile Computing Conference (IWCMC),* pp. 617-622, August 2014.

[20] S. Alcock and R. Nelson, "Libprotoident: Traffic classification using lightweight packet inspection," WAND Network Research Group, Waikato, 2012.

[21] The French Constitutional Council , "Decision n° 2009-580 of June 10th 2009," 10 June 2009. [Online]. Available: http://www.conseil-constitutionnel.fr/conseil-

constitutionnel/root/bank/download/2009-580DC-2009_580dc.pdf. [Accessed 6 September 2014].

[22] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," *International Conference on Passive and Active Network Measurement,* pp. 165-175, April 2007.

[23] C. Bacquet, A. N. Zincir-Heywood et M. I. Heywood, «An investigation of multi-objective genetic algorithms for encrypted traffic identification,» *Computational Intelligence in Security for Information Systems ,* pp. 93-100, September 2009.

[24] R. Alshammari and A. N. Zincir-Heywood, "A preliminary performance comparison of two feature sets for encrypted traffic classification," *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS'08,* pp. 203-210, October 2008.

[25] R. Alshammari and A. N. Zincir-Heywood, "Machine learning based encrypted traffic classification: Identifying ssh and skype," *Computational Intelligence for Security and Defense Applications,* pp. 1-8, July 2009.

[26] R. Bar-Yanai, M. Langberg, D. Peleg and L. Roditty, "Realtime classification for encrypted traffic," *International Symposium on Experimental Algorithms,* pp. 373-385, May 2010.

[27] A. Dainotti, A. Pescapé et C. Sansone, «Early classification of network traffic through multi-classification,» *International Workshop on Traffic Monitoring and Analysis,* pp. 122-135, April 2011.

[28] D. J.-H. A. N. Arndt, "A comparison of three machine learning techniques for encrypted network traffic analysis," *Computational Intelligence for Security and Defense Applications (CISDA) ,* pp. 107-114, April 2011.

[29] Y. Okada, S. Ata, N. Nakamura, Y. Nakahira and I. Oka, "Application identification from encrypted traffic based on characteristic changes by encryption," *Communications Quality and Reliability (CQR),* pp. 1-6, May 2011.

[30] C. Bacquet, A. N. Zincir-Heywood et M. I. Heywood, «Genetic optimization and hierarchical clustering applied to encrypted traffic identification,» *Computational Intelligence in Cyber Security (CICS),* pp. 194-201, April 2011.

[31] Y. Du and R. Zhang, "Design of a method for encrypted P2P traffic identification using K-means algorithm," *Telecommunication Systems,* vol. 53, no. 1, pp. 163-168, 2013.

[32] H. A. H. Ibrahim, O. R. A. Al Zuobi, M. A. Al-Namari, G. MohamedAli and A. A. A. Abdalla, "Internet traffic classification using machine learning approach: Datasets validation issues," *Basic Sciences and Engineering Studies (SGCAC),* pp. 158-166, February 2016.

[33] L. Deri, "nProbe: an open source netflow probe for gigabit networks," *TERENA Networking Conference,* pp. 1-4, 2003 May 2003.

[34] K. P. Gummadi, S. Saroiu and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment,* pp. 5-18, November 2002.

[35] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," *Proceedings of Passive and Active Measurements (PAM) Workshop,* pp. 14-25, March 2002.

[36] Y. Zhang, L. Breslau, V. Paxson and S. Shenker, "On the characteristics and origins of internet flow rates," *ACM SIGCOMM Computer Communication Review,* vol. 32, no. 4, pp. 309-322, 2002.

[37] B. Claise, "Cisco systems netflow services export version 9.," October 2004. [Online]. Available: https://tools.ietf.org/html/rfc3954. [Accessed 2 May 2015].

[38] B. Claise, B. Trammell and P. Aitken, "RFC 7011 - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," September 2013. [Online]. Available: https://tools.ietf.org/html/rfc7011. [Accessed 28 April 2016].

[39] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano and F. Risso, "Gt: picking up the truth from the ground for internet traffic," *ACM SIGCOMM Computer Communication Review,* vol. 39, no. 5, pp. 12-18, 2009.

[40] V. Carela-Español, T. Bujlow and P. Barlet-Ros, "Is our ground-truth for traffic classification reliable?," *International Conference on Passive and Active Network Measurement ,* pp. 98-108, March 2014.

[41] K. L. Calvert, W. K. Edwards, N. Feamster, R. E. Grinter, Y. Deng and X. Zhou, "Instrumenting home networks," *ACM SIGCOMM Computer Communication Review,* vol. 41, no. 1, pp. 84-89, 2011.

[42] J. Whiteaker, F. Schneider, R. Teixeira, C. Diot, A. Soule, F. Picconi and M. May, "Expanding home services with advanced gateways," *ACM SIGCOMM Computer Communication Review,* vol. 42, no. 5, pp. 37-43, 2012.

[43] Université Pierre et Marie Curie, "D2.2 Design document of gateway-centric monitoring tools," 24 May 2013. [Online]. Available: http://cordis.europa.eu/docs/projects/cnect/8/258378/080/deliverables/001-FIGAROD22v11finalwithappendices.pdf. [Accessed 11 January 2017].

[44] S. Hätönen, A. Nyrhinen, L. Eggert, S. Strowes, P. Sarolahti and M. Kojo, "An experimental study of home gateway characteristics," *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement,* pp. 260-266, November 2010.

[45] K. C. Claffy, H. W. Braun and G. C. Polyzos, "A parameterizable methodology for Internet traffic flow profiling," *IEEE Journal on selected areas in communications,* vol. 13, no. 8, pp. 1481-1494, 1995.

[46] M. Kaplow, "FOKUS - Fraunhofer Institute for Open Communication Systems," 24 January 2017. [Online]. Available: https://cdn2.scrvt.com/fokus/cbc8c2f1df9d3f0c/381b99c85752/FOKUS-Infoblatt_A4_170124_e.pdf. [Accessed 28 March 2017].

[47] ETH Zürich, "Network Security Group, ETH Zürich," 2017. [Online]. Available: http://www.netsec.ethz.ch/. [Accessed 12 February 2017].

[48] WAND, "WAND - WAND Research Group," 2017. [Online]. Available: https://wand.net.nz/. [Accessed 12 February 2017].

[49] C. M. Inacio and B. Trammell, "Yaf: yet another flowmeter," *Proceedings of LISA'10: 24th Large Installation System Administration Conference,* pp. 107-115, November 2010.

[50] S. Alcock, P. Lorier and R. Nelson, "Libtrace: a packet capture and analysis library," *ACM SIGCOMM Computer Communication Review,* vol. 42, no. 2, pp. 42-48, 2012.

[51] L. Deri, "Improving passive packet capture: Beyond device polling," *Proceedings of SANE,* pp. 85-93, September 2004.

[52] L. Rizzo, "Netmap: a novel framework for fast packet I/O," *21st USENIX Security Symposium (USENIX Security 12),* pp. 101-112, August 2012.

[53] Intel Corporation, "DPDK - Getting Started Guide," 24 April 2014. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/guides/dpdk-getting-started-guide.pdf. [Accessed 25 July 2016].

[54] A. Dainotti, A. Pescape and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE Network,* vol. 26, no. 1, pp. 36-40, 2012.

[55] J. Micheel, S. Donnelly and I. Graham, "Precision timestamping of network packets," *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement ,* pp. 273-277, November 2001.

[56] T. Zseby, M. Molina, N. Duffield, S. Niccolini and F. Raspall, "RFC5475 - Sampling and Filtering Techniques for IP Packet Selection," March 2009. [Online]. Available: https://tools.ietf.org/html/rfc5475. [Accessed 2 May 2015].

[57] B. Claise, G. Dhandapani, P. Aitken and S. Yates, "RFC 6313 - Export of structured data in IP Flow Information Export (IPFIX)," July 2011. [Online]. Available: https://www.ietf.org/rfc/rfc6313.txt. [Accessed 16 November 2015].

[58] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen and P. Conrad, "RFC 3758 - Stream Control Transmission Protocol (SCTP) Partial Reliability Extension," May 2004. [Online]. Available: https://tools.ietf.org/html/rfc3758. [Accessed 6 June 2014].

[59] J. Fan, J. Xu, M. H. Ammar and S. B. Moon, "Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme," *Computer Networks,* vol. 46, no. 2, pp. 253-272, 2004.

[60] B. Trammell, A. Wagner and B. Claise, "RFC 7015 - Flow aggregation for the IP flow information export protocol," September 2013. [Online]. Available: https://tools.ietf.org/html/rfc7015. [Accessed 12 August 2016].

[61] Center for Applied Internet Data Analysis (CAIDA), "Summary of Anonymization Best Practice Techniques," 5 April 2016. [Online]. Available: http://www.caida.org/projects/predict/anonymization/. [Accessed 25 December 2016].

[62] C. Schmoll, S. Teofili, E. Delzeri, G. Bianchi, I. Gojmerac, E. Hyytia, B. Trammell, E. Boschi, G. Lioudakis, F. Gogoulos, A. Antonakopoulou, D. Kaklamani and I. Venieris, "State of the art on data protection algorithms for monitoring systems. PRISM. IST-2007-215350, Data Protection Algorithms," 30 June 2008. [Online]. Available: http://www.cspforum.eu/fp7-prism-wp3.1-d3.1.1-final.pdf. [Accessed 22 February 2017].

[63] T. Bujlow, K. Balachandran, T. Riaz and J. M. Pedersen, "Volunteer-Based System for classification of traffic in computer networks," *19th Telecommunications Forum (TELFOR),* pp. 210-213, November 2011.

[64] D. Moore, K. Keys, R. Koga, E. Lagache and K. C. Claffy, "The coralreef software suite as a tool for system and network administrators," *Proceedings of the 15th USENIX conference on System administration,* pp. 133-144, December 2001.

[65] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," *International Workshop on Passive and Active Network Measurement,* pp. 41-54, March 2005.

[66] T. Bhatia, "OpenDPI v.3.10," 15 July 2012. [Online]. Available: https://github.com/thomasbhatia/OpenDPI. [Accessed 2 June 2016].

[67] J. Levandoski, E. Sommer and S. M, "Application Layer Packet Classifier for Linux," 7 January 2009. [Online]. Available: http://l7-filter.sourceforge.net/. [Accessed 2014 April 24].

[68] T. Bujlow, V. Carela-Español and P. Barlet-Ros, "Independent comparison of popular DPI tools for traffic classification," *Computer Networks,* vol. 76, no. C, pp. 75-89, 2015.

[69] M. Finsterbusch, C. Richter, E. Rocha, J. A. Muller and K. Hanssgen, "A survey of payload-based traffic classification approaches," *IEEE Communications Surveys & Tutorials,* vol. 16, no. 2, pp. 1135-1156, 2014.

[70] N. Cascarano, A. Este, F. Gringoli, F. Risso and L. Salgarelli, "An experimental evaluation of the computational cost of a DPI traffic classifier," *Global Telecommunications Conference (GLOBECOM),* pp. 1-8, November 2009.

[71] N. Cascarano, L. Ciminiera and F. Risso, "Improving cost and accuracy of DPI traffic classifiers," *Proceedings of the 2010 ACM Symposium on Applied Computing,* pp. 641-646, March 2010.

[72] C. V. Wright, F. Monrose and G. M. Masson, "On inferring application protocol behaviors in encrypted network traffic," *Journal of Machine Learning Research,* vol. 7, no. Dec, pp. 2745-2769, 2006.

[73] Y. Kumano, S. Ata, N. Nakamura, Y. Nakahira and I. Oka, "Towards real-time processing for application identification of encrypted traffic," *Interntional conference on Computing, Networking and Communications (ICNC),* pp. 136-140, February 2014.

[74] T. Bujlow, T. Riaz and J. M. Pedersen, "A method for classification of network traffic based on C5. 0 Machine Learning Algorithm," *International conference on Computing, Networking and Communications (ICNC),* pp. 237-241, January 2012.

[75] V. Carela-Español, P. Barlet-Ros, A. Bifet and K. Fukuda, "A streaming flow-based technique for traffic classification applied to 12+ 1 years of Internet traffic," *Telecommunication Systems,* vol. 63, no. 2, pp. 191-204, 2016.

[76] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials,* vol. 10, no. 4, pp. 56-76, 2008.

[77] P. Velan, M. Čermák, P. Čeleda and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management,* vol. 25, no. 5, pp. 355-374, 2015.

[78] J. Khalife, A. Hajjar and J. Diaz-Verdejo, "A multilevel taxonomy and requirements for an optimal traffic-classification model," *International Journal of Network Management,* vol. 24, no. 2, pp. 101-120, 2014.

[79] A. Moore, D. Zuev and M. Crogan, "Discriminators for use in flow-based classification," August 2005. [Online]. Available: https://qmro.qmul.ac.uk/xmlui/bitstream/handle/123456789/5050/RR-05-13.pdf?sequence=1. [Accessed 7 May 2016].

[80] J. R. Quinlan, C4. 5: programs for machine learning, London: Elsevier, 2014.

[81] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster and Y. Q. Song, "FlowQoS: QoS for the rest of us," *Proceedings of the third workshop on Hot topics in software defined networking,* pp. 207-208, August 2014.

[82] P. Arcaini, E. Riccobene and P. Scandurra, "Modeling and analyzing MAPE-K feedback loops for self-adaptation," *International Symposium on Software Engineering for Adaptive and Self-Managing Systems,* pp. 13-23, May 2015.

[83] L. Deri, "How to Enforce Layer-7 Traffic Policies Using ntopng," 9 February 2015. [Online]. Available: http://www.ntop.org/ndpi/how-to-enforce-layer-7-traffic-policies-using-ntopng/. [Accessed 2 May 2017].

[84] A. Dainotti, W. De Donato and A. Pescapé, "A community-oriented traffic classification platform," *International Workshop on Traffic Monitoring and Analysis,* pp. 64-74, 2009.

[85] "libreProbe source code," [Online]. Available: https://github.com/kvitaly2005/lprobe. [Accessed 4 February 2016].

[86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel and J. Vanderplas, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research (JMLR),* vol. 12, pp. 2825-2830, 2011.

[87] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, " The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter,* pp. 10-18, 2009.

[88] Rulequest Research, «C5.0 DocumentationAvailable,» March 2017. [En ligne]. Available: https://www.rulequest.com/see5-unix.html. [Accès le 2 June 2017].

[89] A. Z. Kortebi, C. Delahaye, J. P. Javaudin and Y. Ghamri-Doudane, "A platform for home network traffic monitoring," *IFIP/IEEE Symposium on Integrated Network and Service Management (IM),* pp. 895-896, May 2017.

[90] ntop, "Why nProbe+JSON+ZMQ instead of native sFlow/NetFlow support in ntopng?," 13 September 2013. [Online]. Available: http://www.ntop.org/nprobe/why-nprobejsonzmq-instead-of-native-sflownetflow-support-in-ntopng/. [Accessed 2 April 2015].

[91] L. Deri, M. Martinelli and A. Cardigliano, "Realtime High-Speed Network Traffic Monitoring Using ntopng," *USENIX Conference on Large Installation System Administration,* 70-80 November 2014.

[92] A. Lahmadi, F. Beck, E. Finickel and O. Festor, "A platform for the analysis and visualization of network flow data of android environments," *IFIP/IEEE International Symposium on Integrated Network Management (IM),* pp. 1129-1130, 2015.

[93] J. Khalife, A. Hajjar and J. Diaz-Verdejo, "A multilevel taxonomy and requirements for an optimal traffic-classification model," *International Journal of Network Management,* vol. 24, no. 2, pp. 101-120, 2014.