Thèse de Doctorat de l'université Paris VI
Pierre et Marie Curie

Spécialité

Systèmes Informatiques

présentée par

Thomas Silverston

pour obtenir le grade de

Docteur de l'université Pierre et Marie Curie

Peer-to-Peer Video Live Streaming: Measurement Experiments and Traffic Analysis

Soutenance prévue le 16 Septembre 2009 devant le jury composé de

MM. :	Laurent	Massoulié	Thomson Lab	Rapporteurs
	Thierry	Turletti	INRIA Sophia Antipolis	
MM. :	Kenjiro	Cho	IIJ Research Lab / WIDE	Examinateurs
	Pierre	Sens	Professeur à l'UPMC Univ Paris 06	
	Ioannis	Stavrakakis	Professor at NKUA (Athens, Greece)	
M. :	Serge	Fdida	Professeur à l'UPMC Univ Paris 06	Directeurs
	Olivier	Fourmaux	MdC à l'UPMC Univ Paris 06	

Thèse de Doctorat de l'université Paris VI
Pierre et Marie Curie

Spécialité

Systèmes Informatiques

présentée par

Thomas Silverston

pour obtenir le grade de

Docteur de l'université Pierre et Marie Curie

Peer-to-Peer Video Live Streaming: Measurement Experiments and Traffic Analysis

Soutenance prévue le 16 Septembre 2009 devant le jury composé de

MM. :	Laurent	Massoulié	Thomson Lab	Rapporteurs
	Thierry	Turletti	INRIA Sophia Antipolis	
MM. :	Kenjiro	Cho	IIJ Research Lab / WIDE	Examinateurs
	Pierre	Sens	Professeur à l'UPMC Univ Paris 06	
	Ioannis	Stavrakakis	Professor at NKUA (Athens, Greece)	
M. :	Serge	Fdida	Professeur à l'UPMC Univ Paris 06	Directeurs
	Olivier	Fourmaux	MdC à l'UPMC Univ Paris 06	

Abstract

The Internet is currently experiencing one of the most important challenges in terms of content distribution since its first uses as a medium for content delivery: users from passive downloaders and browsers are moving towards content producers and publishers. Following this change, the content delivery architecture of the Internet is also evolving from the classical client/server model to the peer-to-peer model (P2P). For new content delivery services that target a large number of receivers on the Internet, there are scalability issues to deliver the content to potentially millions of users at the same time. This is clearly the case for television service on the Internet.

The television broadcast service with a peer-to-peer architecture (P2P-TV) has became an important research topic since it is expected that P2P-TV applications will become massively used on the Internet. There has been a lot of P2P-TV architecture proposals but, none of these proposals has been really implemented or deployed. At the same time, many commercial applications appeared on the Internet (PPLive, SOPCast) and their popularity is increasing. Even tough these applications are freely available, their source code is not open and their exact implementation details and protocols are still widely unknown. There is a lack of knowledge with these applications, with regards to their architectures, mechanisms or traffic.

In this thesis, in order to fill the gap between the architecture proposals and the commercial applications, we study the P2P-TV applications by performing extensive measurement experiments. We passively measured the network traffic generated by popular P2P-TV applications.

We present a detailed study of the P2P-TV traffic, providing useful insights on both transport and packet-level properties as well as on the behavior of the peers inside the network. The knowledge gained thanks to this traffic measurement and analysis is useful for several tasks as traffic identification, understanding the performance of different P2P-TV technologies, the impact of such traffic on the network and building more realistic models for simulations.

In order to extend the scope of our study, we performed another measurement campaign at larger scale between Japan and France to better characterize the P2P network. From these experiments, we study the global organization of peers in the network, the amount of traffic they exchange and the overall collaboration of peers. We will also

focus on the distance between peers and the geographic location of the users.

Keywords: content distribution, peer-to-peer, P2P-TV, measurement experiments, traffic analysis, distributed systems.

Résumé

L'Internet est en train de connaître l'un de ses plus importants bouleversements en termes de distribution de contenu depuis qu'il est utilisé comme support de diffusion : les utilisateurs jusqu'alors passifs récepteurs de contenu deviennent producteurs de contenus. Suite à ce changement, l'architecture de diffusion de contenu de l'Internet évolue également passant du modèle classique client/serveur au modèle pair-à-pair (P2P). Pour les nouveaux services de diffusion de contenu qui visent un grand nombre de récepteurs sur l'Internet, il y a des inquiétudes concernant leur aptitude à transmettre les contenus à large échelle à des millions d'utilisateurs en même temps. C'est typiquement le cas pour le service de diffusion de télévision sur l'Internet.

La diffusion de télévision sur l'Internet via une architecture pair-à-pair (P2P-TV) est devenu un sujet de recherche important puisque l'on s'attend à ce que ces applications soient massivement utilisées sur l'Internet. Ainsi, il y a eu de nombreuses propositions d'architectures P2P-TV mais aucune d'entre elles n'a réellement été implémentée ni même déployée. Dans le même temps, des applications P2P-TV commerciales sont apparues sur l'Internet (PPLive, SOPCast) et sont devenues de plus en plus populaire. Bien que ces applications soient disponibles sur l'Internet, elles sont propriétaires et leur code source n'est pas ouvert : les détails d'implémentation ainsi que les protocoles qu'elles utilisent sont complètement inconnus. Il y a un manque total de connaissance de ces applications, aussi bien d'un point de vue architectural, que du point de vue de leurs mécanismes ou du trafic qu'elles génèrent.

Dans cette thèse, afin de combler l'écart entre les propositions d'architectures et les applications commerciales, nous avons étudié les applications P2P-TV en effectuant d'importantes expériences de mesures. Nous avons mesuré passivement le trafic réseau généré par les applications P2P-TV les plus populaires.

Nous présentons donc une étude détaillée du trafic P2P-TV, fournissant des résultats utiles à la fois sur les propriétés de niveau paquet et transport du trafic que sur le comportement des pairs dans le réseau. Ces expériences de mesures et analyses de trafic sont utiles pour comprendre les technologies P2P-TV utilisées, identifier ce type de trafic, évaluer son impact sur le réseau et permettre l'élaboration de modèles plus réalistes pour les simulations.

Afin d'étendre la portée de notre étude, nous avons effectué une seconde campagne

de mesure à plus large échelle entre le Japon et la France afin de mieux caractériser les réseaux P2P-TV mesurés. Grâce à cette nouvelle expérience, nous étudions l'organisation globale des pairs dans le réseau, la quantité de trafic échangé et le niveau de collaboration des pairs. Nous nous intéressons également à la distance entre les pairs et à la situation géographique des utilisateurs.

Mots-clés : distribution de contenu, pair-à-pair, P2P-TV, expériences de mesure, analyse de trafic, systèmes distribués.

Contents

1 Introduction **1**
- 1.1 Challenges . . . 2
- 1.2 Thesis Contribution . . . 3
- 1.3 Document Structure . . . 4

2 Architectures for P2P Live Streaming: State of the Art **5**
- 2.1 Tree-based Architectures . . . 9
  - 2.1.1 Single-tree Approach . . . 9
    - 2.1.1.1 Peercast . . . 10
    - 2.1.1.2 NICE . . . 11
    - 2.1.1.3 Narada . . . 12
    - 2.1.1.4 Bullet . . . 13
  - 2.1.2 Multiple-trees Approach . . . 14
    - 2.1.2.1 Splitstream . . . 15
    - 2.1.2.2 CoopNet . . . 16
    - 2.1.2.3 PALS . . . 17
- 2.2 Mesh-based Architectures . . . 17
  - 2.2.1 BitTorrent . . . 19
  - 2.2.2 DONet . . . 21
  - 2.2.3 GridMedia . . . 22
  - 2.2.4 Pulse . . . 23
  - 2.2.5 R^2 . . . 24
  - 2.2.6 PRIME . . . 25
  - 2.2.7 Chunkyspread . . . 26
- 2.3 Summary . . . 28
- 2.4 Discussion . . . 29

3 Peer-Centric Experiment of P2P-TV **31**
- 3.1 Description of the Experiment . . . 32
  - 3.1.1 Studied P2P-TV Applications . . . 32
  - 3.1.2 Measurement Experiments Testbed . . . 33

3.2 Understanding P2P-TV Traffic . . . 35
3.2.1 Transport Protocols . . . 36
3.2.2 Ports . . . 38
3.3 Signaling and Video Traffic . . . 39
3.4 Scaling Behavior of the Traffic . . . 42
3.4.1 Differences between TCP and UDP Traffic . . . 43
3.4.2 Impact of the Signaling Traffic . . . 45
3.5 Understanding P2P-TV Peer Behavior . . . 48
3.5.1 Traffic Generated and Received by Peers . . . 48
3.5.2 Peers Neighborhood and Session Duration . . . 51
3.6 Summary . . . 55

4 P2P-TV Experiment on Both Sides of The World 57
4.1 Motivation . . . 58
4.2 Related Measurement Experiment of P2P-TV . . . 59
4.2.1 Measurement with Global Knowledge of the System . . . 60
4.2.2 Active Measurements of P2P-TV . . . 62
4.2.3 Passive Measurements of P2P-TV . . . 63
4.2.4 Discussion . . . 64
4.3 Presentation of the Experiment . . . 65
4.3.1 Large-Scale Experimental Testbed . . . 65
4.3.2 P2P-TV Systems Under Test . . . 66
4.3.3 Process of Experiment . . . 67
4.4 General Observations . . . 68
4.4.1 Similarity of Peers in the Traces . . . 71
4.5 Exchange of Data between Peers . . . 74
4.6 Distance between Peers . . . 76
4.6.1 Number of hops . . . 77
4.6.2 Autonomous Systems and Countries . . . 80
4.7 Geographic Location of Peers . . . 84
4.7.1 Influence of the Content on the Population . . . 85
4.7.2 Number of Peers vs. Volume of Traffic . . . 86
4.7.3 Traffic in Upload and Download . . . 87
4.7.4 Stability of the Traffic . . . 94
4.7.5 Discussion . . . 95
4.8 From Measurements to a New Incentive Mechanism . . . 98
4.8.1 Towards a Novel Incentive Mechanism for P2P multimedia live streaming . . . 99
4.9 Summary . . . 101

5 Conclusion and Perspectives 105

A Thesis Publications 109

B Version française **111**
B.1 Introduction . . . 111
B.1.1 Challenges . . . 112
B.1.2 Contribution de la thèse . . . 114
B.1.3 Structure du document . . . 115
B.2 Architectures P2P de diffusion de flots multimédia en direct : état de l'art 116
B.2.1 Architectures construisant un arbre . . . 120
B.2.1.1 Approche à arbre unique . . . 120
B.2.1.2 Approche à arbres multiples . . . 121
B.2.2 Architectures construisant un maillage . . . 124
B.2.3 Discussion . . . 125
B.3 Expériences de mesures P2P-TV locales . . . 128
B.3.1 Expériences de mesures . . . 128
B.3.1.1 Plateforme expérimentale de mesures . . . 128
B.3.2 Étude du trafic P2P-TV . . . 129
B.3.2.1 Protocoles de transport . . . 130
B.3.2.2 Ports . . . 132
B.3.3 Trafic de signalisation et vidéo . . . 133
B.3.4 Comportement d'échelle du trafic P2P-TV . . . 133
B.3.4.1 Différences entre trafic TCP et UDP . . . 134
B.3.4.2 Impact du trafic de signalisation . . . 134
B.3.5 Comportement des pairs P2P-TV . . . 135
B.3.5.1 Trafic généré et reçu par les pairs . . . 135
B.3.5.2 Voisinage des pairs et durée des sessions . . . 137
B.4 Mesures d'applications P2P-TV de part et d'autre du monde . . . 140
B.4.1 Expérience de mesures large-échelle . . . 140
B.4.1.1 Motivation . . . 140
B.4.1.2 Plate-forme expérimentale de mesures . . . 141
B.4.1.3 Déroulement des expériences . . . 141
B.4.2 Observations générales . . . 142
B.4.3 Échange de trafic entre les pairs . . . 145
B.4.4 Distance entre les pairs . . . 146
B.4.4.1 Systèmes autonomes . . . 148
B.4.4.2 Localité géographique . . . 149
B.5 Conclusion et Perspectives . . . 152

Bibliography **154**

List of Figures **164**

List of Tables **167**

Chapter 1

Introduction

In recent years we have experienced a dramatically growing impact of the users influence on the evolution of the Internet and its services. Users create events, making new contents and services available; they create communities, in which active participation, user interaction, and information sharing are highly encouraged; and require new technologies to support them. User demands and new forms of interaction drive the network evolution, pushing for new network applications, new communication paradigms, and new network architectures. A few notable examples of this small revolution are the explosion of Internet Blogs, video publishing and distribution systems, social networks built through the Web, virtual worlds, network games, and many others [1, 2].

Therefore, by interacting through the network, users create new forms of communities and new forms of content distribution: we are witnessing to a shift from the traditional distribution paradigm of a few content providers with many consumers, to a new paradigm that sees many content providers and as many consumers [3].

In addition, the availability of new services and forms of interaction driven by users are, at the same time, changing users' behaviors and expectations. People start using the Internet for activities previously happening only in certain contexts and through different technologies. This is the case, for example, of Internet Television (IPTV), and network gaming in virtual worlds. The time and place of such activities have changed, and services become ubiquitous. People move from the *sofa at home* to the workplace or a café to enjoy such services. Moreover, they interact with communities that range on a global scale rather than having a strong local geographical bound. Such new scenarios make the traditional content distribution systems partially dated, thus increasing the interest of network operators and industry in general to support new service typologies. This phenomenon is making all the current content distribution networks (Television, Telephony, Data) converge on a single one: the Internet.

The popularity of Internet-based television is expected to grow during the next decades for several reasons [4]. First, it is well known that in the recent past, especially for some events such as the 9/11, the Internet has been the major source of information

for people at their workplace. Second, users appreciate the generalist television (TV) always less, whereas they are more interested in specialized content on TV and in being able to interact somehow with other users or by adding content (commenting or asking questions to the community watching the same videos is one of the simplest examples) [5]. Third, in some countries the quality and the range of the offer of traditional television contents are scarce. Finally, as fourth, the "*Broadcast yourself*" phenomenon is constantly increasing, both with "**Tube*" sites and the creation of more elaborate personal TV programs with real-time broadcasting created by single users [6–8]. To testify such trends, several sources report on the loss of audience and of monetary income of the traditional TV and advertisement industries. Therefore, the interest in understanding such new technologies to support and improve them is enormous [9].

The use of the Peer-to-Peer paradigm to deliver live television on the Internet (P2P-TV) is gaining increasing attention [10], and has become a promising alternative to other legitimate approaches as the classical client-server model, Content Delivery Networks [11], or IP-Multicast. Moreover, by using the existing Internet infrastructure as a medium and by exploiting users' participation for the creation of the content distribution network, P2P-TV technologies have innovative potentials: (i) to make any TV channel from any country available worldwide, (ii) to make each Internet user a content creator and distributor by broadcasting his own "TV" with negligible costs.

These are some of the reasons behind the increasing popularity of such applications among Internet users. All these emerging applications will raise new challenges that we will present in the following section.

1.1 Challenges

In this work dissertation, we focus on the P2P-TV systems. More precisely, we performed measurement experiments of these systems and analyzed the network traffic in order to get knowledge of these systems.

At the beginning of this thesis, we noticed that many P2P architectures have already been proposed to stream live multimedia flows. Only a very few of them have been truly implemented or experimented [12, 13], but only at a limited scale. Most of these architecture proposals rely on simulations to validate the proposed architecture and mechanisms. In the meantime, many commercial applications were released on the Internet [14–20] without providing any detail about their architecture and implementation. Thus, there was no knowledge about these systems and no information was available to evaluate them properly. In these conditions, accurate simulation of these systems was very tricky because of the lack of realistic parameters, scenarios, or even simulation tools adapted to the flowing nature of the content. Furthermore, a precise knowledge of these systems could help to uncover the weaknesses of P2P-TV that need further investigations.

Thus, we adopt an opposite approach to lead our research. Instead of focusing on the design of new architectures or mechanisms regardless of their feasibility on the Internet, we focus on the commercial applications that have already been successfully

deployed on the Internet. Our first objective is to get a better understanding of P2P-TV systems. Our observations and findings shall allow to evaluate these systems, to provide information for future realistic models or simulations and to propose new paradigms or mechanisms to improve them. Our approach follows "in vivo" observations of real systems on the Internet, and not "in vitro" assumptions for new architecture proposals.

Since the commercial applications are proprietary and their source code is not open, we can only get knowledge of these systems by measuring them to infer their underlying mechanisms. The only feasible way is to collect and analyze their network traffic. To this end, we performed extensive traffic measurement experiments of commercial P2P-TV applications. This external approach, considering the applications as "black boxes", is a first step to fill the gap between "academic" proposals and "commercial" applications.

Besides, the popularity of P2P-TV is increasing and counts already millions of users all around the world. With the success of P2P file-sharing applications (BitTorrent, eDonkey [21, 22]) or video services (Youtube, Dailymotion [23, 24]) the P2P traffic and video traffic count already for the major part of the overall Internet traffic [25]. It is therefore expected that P2P-TV will contribute to increase the Internet traffic and count for a significant part of this traffic. It is also intuitive that the behavior of users watching television will have an effect on the generated traffic. For instance, users may behave as with traditional TV and often switch from one channel to the other (*channel zapping*).

1.2 Thesis Contribution

In this thesis we focus our research on the study of P2P-TV systems. More precisely, we perform measurement experiment and study the network traffic generated by popular P2P-TV applications at the time of the experiment, and still considered today among the top P2P-TV applications: PPLive, PPStream, Sopcast and TVAnts. Analyzing different applications makes our analysis more complete and allows investigating the generalization of the observed results.

In a first step, we aim at a better understanding of the mechanisms used by such applications and their impact on the network, despite their use of proprietary unpublished protocols, by directly looking at the traffic they generate. This work relies on the detailed analysis of the network traffic generated by the applications in our campus network. First, we identify signaling and video-streaming flows, and examine statistical properties of the produced traffic. We also divided the traffic into upload and download directions to study and characterize them separately. We inspect which transport-level protocols are used and what are the consequences of different choices. Some of the results show how this traffic is quite different from other network applications traffic, and how it may be hard to identify. Moreover they give insights on the characteristics of this traffic to understand its impact on network. Second, we move our analysis to study the behavior of the peers, how peers interact, how much they contribute to the

content distribution and what is the download policy of the different applications. We then observe the phenomenon of peer churn and what is their typical lifetime.

Thanks to such twofold analysis we pave the way to the understanding of both the traffic characteristics of these novel applications and of their related behavior of user. The results presented here are relevant to identify traffic generated by such applications, to understand their impact on network, and to build realistic simulations.

Following our first fruitful measurement campaign, we wanted to extend the scope of our study. Our first experiment focus on the local properties of the P2P-TV systems such as the characteristics of their traffic or their underlying mechanisms. Nowadays, we aim to characterize the P2P network, and the organization of peers in the overall system.

We then perform a second measurement campaign at a larger scale in order to obtain a more global view of the P2P networks. Instead of measuring P2P-TV systems with a single measurement point, we measured them with multiple measurement points located on both sides of the world: in France and in Japan. Based on the measurement of this second experiment, we will discuss mainly about the level of collaboration between peers and the locality of the traffic. Indeed, the P2P traffic has become an important concern for ISPs to perform traffic engineering and to provision the resources of their network. The reciprocal exchange of data between peers is also an essential condition for the scalability of the P2P systems. Moreover, among interesting topics, we study the global organization of peers in the network, the amount of traffic they exchange and the overall collaboration of peers. We will also focus on the distance between peers, the geographic location of the users and the effect of the content on the P2P-TV communities.

1.3 Document Structure

This thesis is divided into five chapters and is structured as follows. In chapter 2, we first give an overview on the techniques to deliver video live streaming on the Internet and survey the main P2P video live streaming architectures. In chapter 3, we present our P2P-TV measurement experiment and traffic analysis. We focus on the characteristics of the generated traffic and we aim at the better understanding of the mechanisms used by such applications and the users' behavior. In order to extend the scope of our measurement experiment, the chapter 4 presents our second measurement experiment at larger scale that has been conducted to better characterize the overall P2P network. Chapter 5 concludes this thesis by summarizing these works and presenting the perspectives.

Chapter 2

Architectures for P2P Live Streaming: State of the Art

From the distribution of content such as web objects or files to multimedia streaming, the content distribution services of the Internet are evolving, leaded by the needs of the Internet customers and the support of the technology. Streaming is often thought of as the playback of continuous flowing media such as Video and Audio [26]. Differently from files which need to be entirely downloaded prior to their playing time, the multimedia streaming flows are played immediately while they are being received to allow to the users to enjoy the content immediately.

By streaming, one may also distinguish live streaming such as television from Video-on-Demand service (VoD). Vod and live video are both streaming services but for VoD, the streamed content is a pre-stored file at the request of the user (movie, TV show) while live streaming broadcasts a content that exhibits an interest to be watched in live (sport events, TV channel, concerts) and which is not a-priori stored on a server. This difference leads to a numerous properties that are specifics to these two distinct streaming services (i.e. playout position of users, startup delay, storage, data persistence, forward functionalities etc.) In this work, we focus on the live streaming service, as this is the case with television.

Video live streaming applications like Television services (Internet-TV) are currently emerging applications on the Internet and are expected to be massively used in the future. Similarly to the classical broadcast television services, these applications target a lot of users at the same time. From the classical client/server architecture commonly used on the Internet to the Peer-to-Peer architectures (P2P), there have been several proposals to allow video live streaming services on the Internet. We can also mention content distribution networks or IP-multicast, which have been considered for this purpose. Currently, the P2P architectures are admitted as being the most suited

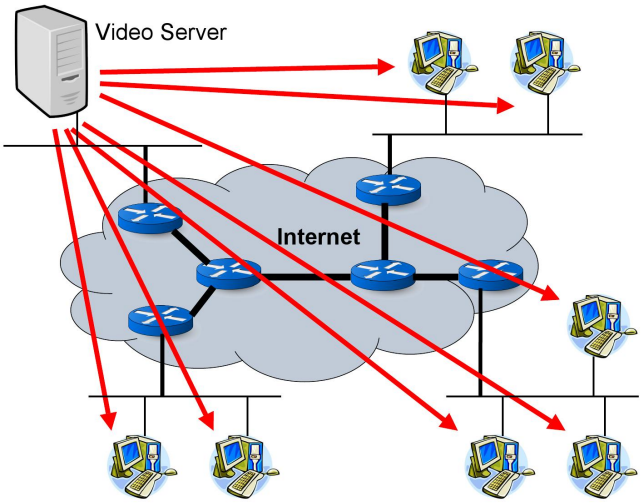


Figure 2.1: Client-server

solutions to enable live streaming on the Internet because the P2P model allows to scale to a large audience, as expected for a television service.

As a beginning of this chapter, we will firstly present the other proposals and explain why they are not adapted to support a television service on the Internet. Then, we will present in details the P2P architectures for video live streaming.

A simple client/server solution where the source server is responsible to duplicate the content with unicast communications to all the receivers will not scale to the large audience expected by the TV broadcasters. This is due to the fact that servers have limited available resources (CPU, bandwidth) that will decrease proportionally with the number of users. This is illustrated on figure 2.1, and it will lead to the inherent bandwidth bottleneck problem on the server-side. When the server achieved its limits, it is not possible for it to serve newcomers. This solution does not scale to the number of users.

The use of Content Distribution Network [11] servers (CDN) could extend the client/server model. In the CDN solutions, as shown on figure 2.2, the video source server pushes the video to CDN servers strategically located close to the users at the network edges. The clients do not download the video from the source but are redirected to a nearby CDN server to download the video. An advantage of the CDN is to shift the traffic load from the core network to the edge. The use of CDN shortens the delay to the users. However, with a large number of users, many CDN servers need to be deployed and it leads to a very expensive infrastructure. The bandwidth provision, at video source servers or in CDNs, must grow proportionally with the client population. For instance, a video platform website like Youtube [23] spend millions dollars a year for bandwidth to perform its video services or to store the video content [27]. We can imagine that it would be more expensive for high quality video services. By

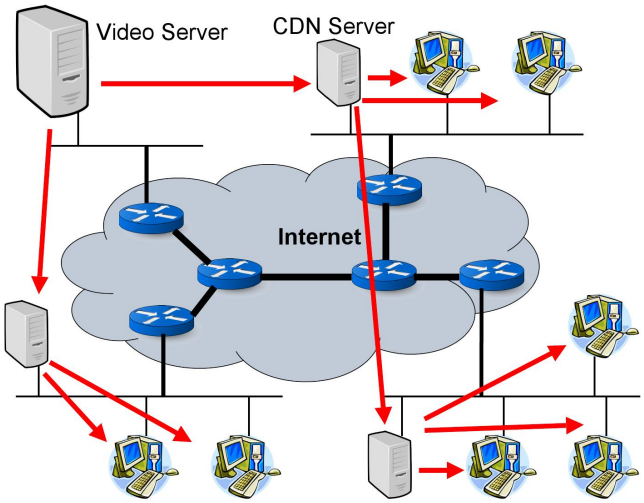


Figure 2.2: CDN servers

multiplying the servers, CDNs only scale to a larger audience with regards to the number of deployed servers. The source server and CDNs would suffer from the inherent bandwidth bottleneck problem with the growing population. Moreover, with live video service, it is a challenging issue to keep updated all the CDN servers with the source server. CDNs have also a high infrastructure cost, which will partially limit its use by the content providers. At the end, the CDN solution is limited in scalability and very expensive (infrastructure, bandwidth).

The live video services could therefore beneficiate of group communication functionalities. However, the architecture of the Internet has been designed for unicast communications (one-to-one) and not for multicast communications (one-to-many). The IP-multicast has been proposed to add group communication functionalities on the Internet [28]. In this context, the Internet routers are responsible for duplicating the packets to each destination and keep only one packet per link, reducing the traffic for all the destinations. This is illustrated on figure 2.3. Unfortunately, the IP-multicast deployment remains limited at the Internet scale because of several complex reasons, which involve not only technical obstacles, but also economic and political concerns [29]. To mention a few of them, the IP-multicast needs changes at the infrastructure level (routers) and this slows down its deployment. These changes would have a very expensive infrastructure cost for the ISPs, to enable all their routers to support the multicast functionalities. The IP-multicast also presents a lot of management concerns such as inter-ISPs traffic or security (DDoS). Finally, there has not been any business model for ISPs to carry multicast traffic and modify their infrastructure. Thus, the lack of deployment of IP-Multicast limits the availability and scope of this approach for a TV service on the Internet scale.

In recent years, with the success of P2P technology for file-sharing [21, 22, 30] or

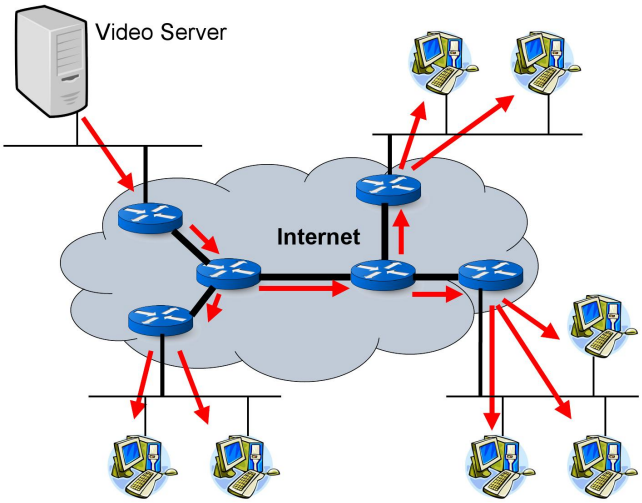


Figure 2.3: Multicast

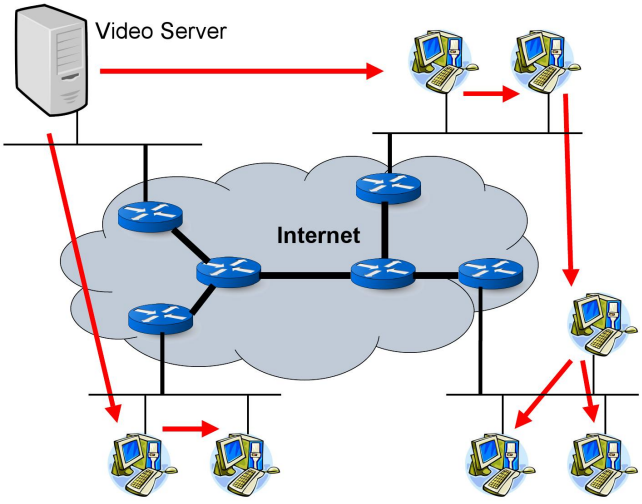


Figure 2.4: Peer-to-Peer

telephony applications [31], there has been significant interest in the use of peer-to-peer paradigm for live video broadcast on the Internet. The key concept of P2P is to encourage the participants to be both clients and servers. The peer-to-peer technology does not require support from Internet routers and network infrastructure and, consequently, is cost-effective and easy to deploy. In P2P networks, the source of the content is not the only one to upload the video streams and to share its resources (bandwidth,

CPU). All the participants, in our case the end-users (also called peers), download the video but also upload it to other peers, sharing their resources to help others to get the data. This is illustrated on the figure 2.4. Consequently, the available resources to deliver the content increase with the number of users and can scale to a large user population.

There has been a lot of proposals to deliver video live streaming over P2P networks. These P2P live streaming architectures can be broadly classified into two categories: tree-based and mesh-based. In the tree-based category, one can also distinguish single-tree or multiple-trees approaches. In fact, this classification follows a chronological order. The single-tree approach was the earlier proposed and only provides group-communications capabilities to stream the live video to a group of peers. Then, the multiple-trees approach was introduced to overcome the limitations of the single-tree approach. Recently, the mesh-based approach appeared and is currently the architecture of reference to design P2P-TV applications. This classification put into perspectives the evolutions of the proposals and presents the challenging issues to design efficient architectures for P2P live streaming applications.

In this chapter, we will present the main architecture proposals for P2P live streaming. We will present each category and illustrate it with representative architectures.

2.1 Tree-based Architectures

As mentioned before, IP-multicast was proposed to add group communication functionalities on the Internet but suffers of a lack of deployment. In an IP-multicast communication, the video source is connected to all the participating users by a multicast tree formed by IP routers in the network. Instead, the group communication functionalities have been implemented in the host of the network, at the application layer. The video source and the receivers form an application level overlay networks to distribute the video content.

2.1.1 Single-tree Approach

Similar to IP-multicast, the data delivery plane is a tree among all the receiver peers rooted at the video source, with each data packet being disseminated using the same structure. A control plane is used to manage the group membership and to help new peers to join the data plane and get the content. The tree-based approach has a well organized overlay structure and the nodes in such a tree have well defined *parent-child* relationships. The peers distribute video by pushing the data to its children peers.

The tree-based approach is illustrated on the figure 2.5 and we present some representative architectures in the rest of the section. We selected these architectures because they all use a different control plane to manage the group membership, on top of which a tree rooted at the source is built to deliver the data among all the receivers (data plane).

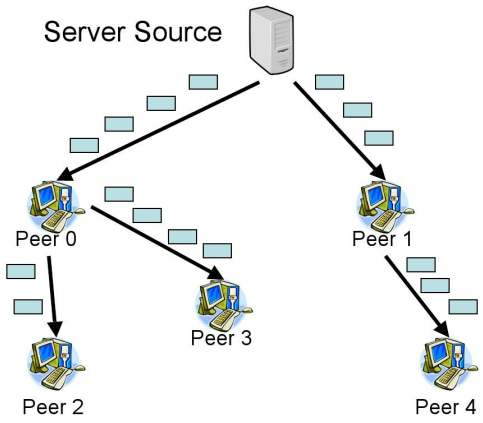


Figure 2.5: Single-tree approach: the data plane is a broadcast tree between all the peers.

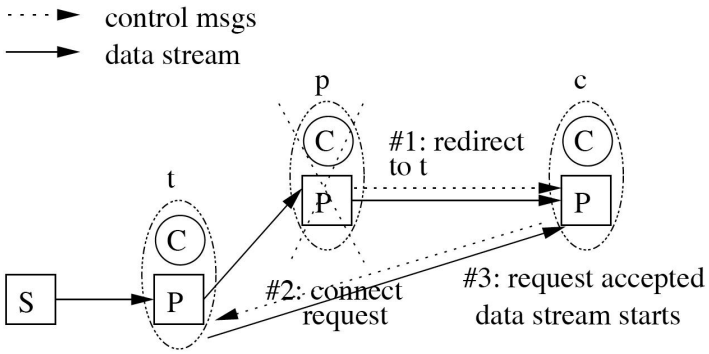


Figure 2.6: Peercast redirection function [32].

2.1.1.1 Peercast

Peercast [32] is one of the first architecture proposal and is also one of the most simple example. For both its data and control planes, Peercast uses the same single tree, which is rooted at the source of the data. The source pushes the data to the peers that are directly connected on it (i.e. its children peers), and it redirects the new joining peers to its children when it has not any available resources. A new joining peer gets across the entire tree structure until it finds another peer able to accept it as a child and to provide it the data. This architecture is very easy to implement since it is just relying on a redirection function between peers, as shown on the figure 2.6. However, it does not provide any additional functionality that will help to maintain the tree in case of failures of peers. If a peer higher in the tree leaves gracefully the network or suffers

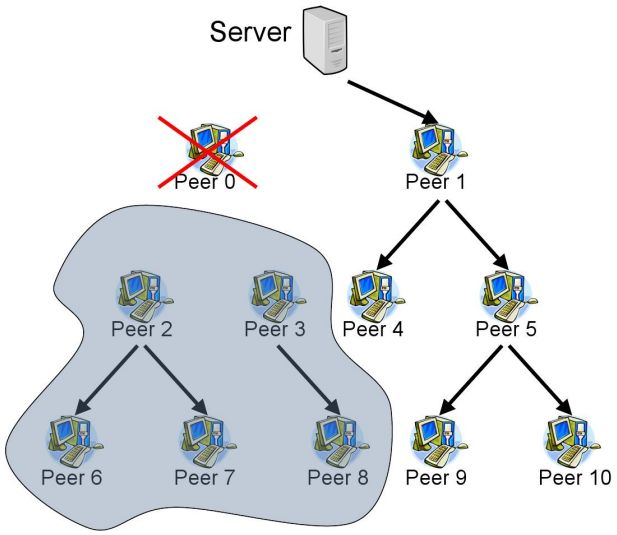


Figure 2.7: Impact of the departure of a peer.

failures, all the lower peers will also be disconnected from the network as it is illustrated on the figure 2.7. The disconnected peers will have to restart a joining procedure from the source. This can take a long time and lead to large packet losses. Moreover, the reconnection procedure generates a lot of signaling packets throughout the entire tree structure, which can prevent Peercast from scaling to large-size networks in the case of massive peers' failures.

In order to deal with this problem, Bawa et al. proposed new policies to reconnect the remaining peers into the tree structure [33]. Only the root of the sub-tree reconnects from the root or from its grandfather in case of a peer departure (i.e. the father of the leaving peer). From the figure 2.7, once Peer 0 leaves or crashes, there are two sub-trees and the roots are Peer 2 and Peer 3 respectively. Only Peer 2 and Peer 3 reconnect from the source, and the other peer of the sub-tree relies on them to recover the data. It alleviates the number of control messages to reconnect all the peers in the tree structure and allow a better scalability for Peercast.

2.1.1.2 NICE

Differently from Peercast, the NICE [34] architecture clearly separates the control and data planes. For its control plane, NICE manages the peers into a set of clusters presented on the figure 2.8. Each cluster has a leader, which is responsible for the group membership within its cluster. The data plane is a still tree, which is built

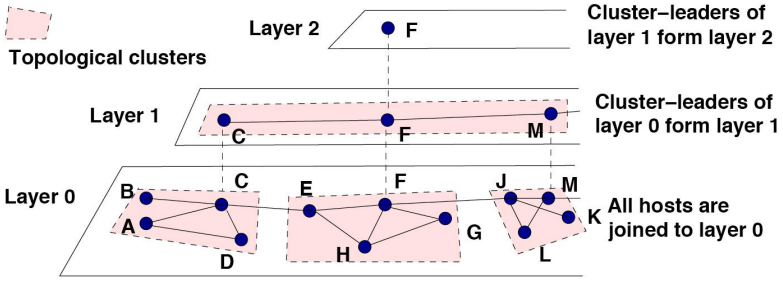


Figure 2.8: NICE: hierarchical set of clusters [34]

from the source among all the cluster leaders. When a new peer joins the network, it contacts the source and it is redirected towards the closest cluster, according to its proximity in delay. When a peer finds a cluster, it contacts the cluster leader and joins this cluster if there are enough resources to provide it data. Otherwise, the peer is redirected towards another cluster until it reaches one able to accept it as a cluster member. Within each cluster, the cluster leader is responsible for delivering the data to the other cluster members.

This architecture has two levels of hierarchy: the clusters' leader and the clusters' members. The division of the control plane into a set of clusters will confine the messages within each cluster. It will alleviate the number of signaling messages throughout the entire control plane when a new peer joins the network. Moreover, only the members of a cluster will suffer from the failure of a leader.

However, the implementation of the mechanisms to gather the peers into clusters could be a very challenging task. Some functionalities like cluster *merge* or *split* may generate a large amount of control messages. Finally, on one hand NICE is able to better mitigate the effect of a pair's failure and to reduce the amount of signaling messages. On the other hand, the complexity to implement such architecture could generate a large amount of signaling messages during the building phase, leading to other scalability concerns. Moreover, the cluster leaders are expected to share a lot of resources while the other cluster members do not use their resources to help to deliver the data. The resources of overall peers are underutilized and could lead to poor performance in the network.

Another similar hierarchical architecture to NICE is ZigZag [35], where the peers do not get the content from their cluster leader but from a foreign cluster leader. ZigZag balances better the load of the cluster leaders compared to Nice. It also keeps a low signaling overhead when inherent peers failure occurs.

2.1.1.3 Narada

In Narada [36], the control plane organize the peers into a mesh that connects all the participating nodes, selecting the links based on round-trip time estimates between nodes. On top of the mesh, as it is illustrated on figure 2.9, Narada builds a source-rooted minimum delay tree used for the data delivery to all the peers.

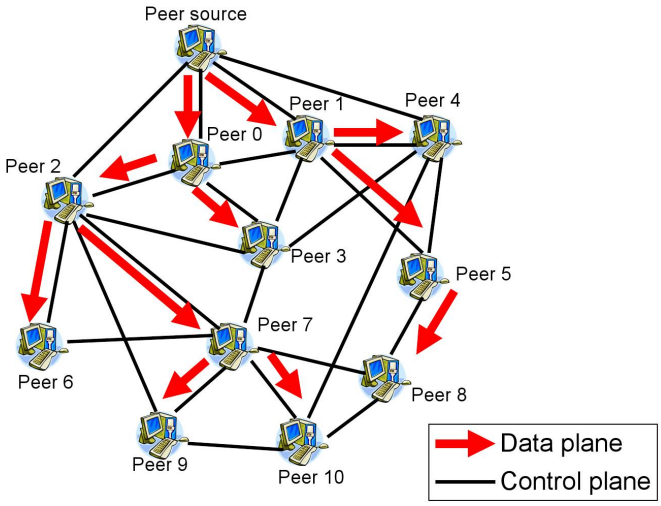


Figure 2.9: Narada: a data delivery tree is built on top of a mesh between peers.

A mesh makes it possible to have more than one path between the source and the receiver peers. When a failure occurs, the effects are limited because the peers already know other paths to recover the data. However, the cost of maintaining a mesh topology is high in term of signaling messages exchanged between peers. Since video live streaming service aims to scale to a large audience, such topology may quickly exhibits major scalability concerns for very large-scale network

2.1.1.4 Bullet

Bullet [37] tries to take advantage of the tree and the mesh structures for its control plane. In Bullet, a mesh is built over the distribution tree and allows participants to recover the data from other peers that only their parent in the tree. The source splits the data and delivers it into distinct sub-trees. In each sub-tree, the peers have to obtain the missing data by creating new links across the other sub-trees, building a mesh on top of the distinct sub-trees (figure 2.10). By mixing a mesh and a tree structures for its control plane, Bullet is able to deal with the failure of peers, which is a major concern with the single-tree structure. Bullet is also able to mitigate the scalability concern of the mesh structure and its high cost in signaling messages.

By separating the data plane into distinct trees, this protocol is an exception in our "single-tree" classification. We choose to keep Bullet into single-tree based architecture because it does not exactly build multiple-trees, as we will show it in the next section. However, the Bullet architecture is a first step towards the multiple-trees approach.

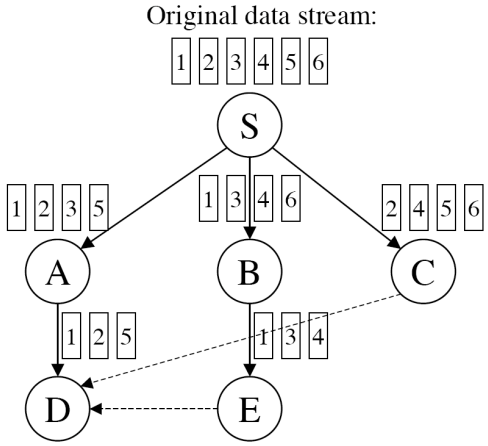


Figure 2.10: Architecture of Bullet [37]: a mesh is built on top of an overlay tree.

2.1.2 Multiple-trees Approach

The single-tree based architectures were the first proposals for streaming live media over P2P networks. At the application-layer, they organize strictly peers into a tree as it was proposed with the IP-multicast. The single-tree approach suffers from two main limitations: (i) the vast majority of lead nodes and under-utilized bandwidth resources; (ii) disruptive delivery due to failures of high-level nodes.

In the former one, the leaves nodes are only receiving data and have no child to forward the data. If a large amount of nodes are leaf nodes, a large amount of resources are not used and it may be an important issue for the scalability of the single-tree approach.

In the latter one, failure or departure of nodes, particularly those higher in the tree, may disrupt delivery of data to a large number of their offspring in the tree (figure 2.7). This phenomenon is more generally referred as the *churn of peers*. The churn of peers is the arrivals and departures of peers in the network at any time.

For the P2P technology we are focusing on, the peers are the end-users, which are not entities dedicated to stay in the network all the time, such as the routers in the Internet. The end-users are prone to failures, and the behavior of users may impact the peers' lifetime in the network. In case of failures, a peer departure will temporarily disrupt the video delivery to all the peers in the sub-tree rooted at the departure peers and they would suffer packet losses. The churn of peers is a very important problem to take into account with P2P networks, especially for video live streaming delivery because of the continuous nature of the broadcasted content and its time-sensitive nature.

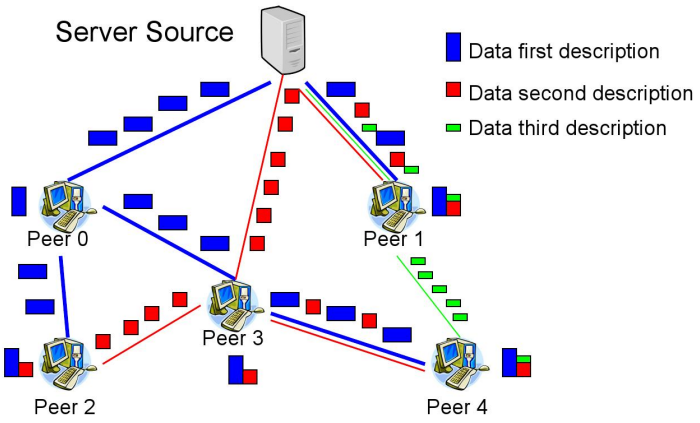


Figure 2.11: Second-generation protocols: the data plan is composed with multiple-trees between all the peers.

To overcome the limitations of the single-tree based approach, more resilient structures such as multiple-trees have been introduced. Here, the source encodes the video stream into sub-streams by using layered encoding [38] (LC) or multiple description coding schemes [39] (MDC). The source distributes each substream along a particular sub-tree (figure 2.11). The quality experienced by a receiver depends on the number of sub-streams it receives. A peer can choose to receive the data from all the trees to obtain the full video quality, or to select only a subset of trees to adapt the quality of the video to its need and capabilities. This scheme allows also the peers with low resources to upload only some parts of the data and not the entire video. It allows to adapt to the heterogeneous capacities of the peers.

There are key advantages of the multiple-trees over the single-tree. With multiple-trees, the potential bandwidth of all nodes can be utilized, as long as each node is not a leaf in at least one tree. A peer, which is a leaf node in a sub-tree, should be also an internal node in another sub-tree, enabling this peer to share its resources to the other peers. In the case of a peer's departure, the overall resiliency of the system is improved, as the children peers are not completely disrupted by the failure of their ancestor on a given tree and continue to receive the data from other sub-trees.

The rest of the section presents the main multiple-trees architecture for P2P live video streaming.

2.1.2.1 Splitstream

Splitstream [40] tackles the problem of the single-tree approach, where leaf nodes do not contribute with their resources for the other peers. To this end, Splitstream delivers the data to the peers by building multiple-trees, where each peer is an interior node in at

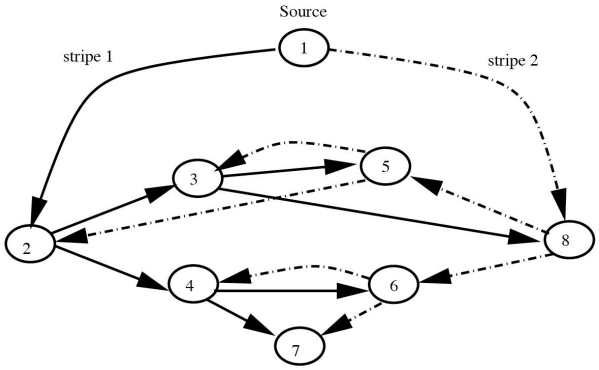


Figure 2.12: Architecture of Splitstream [40]. An independent tree is built for each stripe.

most one tree, and a leaf node in the other trees (figure 2.12). To achieve this property, the trees are node-disjoint. Each tree is built by using Scribe [41], an application-layer group communication system. It relies on Pastry [42], a Distributed Hash Table protocol (DHT) used to locate parents and assign positions of nodes in each tree. The data are divided into stripes, and are propagated using a different tree per stripe. A receiver aiming to improve its received quality, choose the number of stripes it need to receive and joins the trees that corresponds to those stripes. In case of a peer failure, it causes only the loss of one stripe for its offspring peers of the sub-tree and limits the decrease in quality.

By using an appropriate data encoding scheme such MDC or LC, the effects of a failure are mitigated, since the video can be reconstituted from any subset of the stripes, and the video quality will depend on the number of received stripes. If an interior node in a sub-tree fails, then the depending nodes are able to continue displaying the media stream at reduced quality until the stripe tree is repaired.

2.1.2.2 CoopNet

CoopNet [43] chose to encode the data with multiple description coding scheme (MDC). All descriptions have the same importance and the quality of reception depends on the number of descriptions received. Each description of the data is independent and is pushed into a specific tree. A peer can join any tree to download a data description and decode the video. The more description a peer downloads, the better quality it experiments. Figure 2.11 illustrates this scheme if you consider every description of the data are equivalent (blue, red and green descriptions).

Differently from Splitstream which is fully distributed, CoopNet uses a central server to assign a peer in the trees structure. The central server ensures that a peer is at least internal node in one sub-tree to contribute with its resources.

2.1.2.3 PALS

In PALS [44], a receiver peer orchestrates the adaptive delivery of data from multiple parent peers. The data are encoded by using layered encoding scheme (LC) and are delivered into multiple-tree to receiver peers. With layered encoding scheme, there is dependency between the layers, composed of a "base layer" and several "enhancement layers" [45]. Each layer of the data is delivered by a different tree. A peer needs to get at least the base layer to decode the video, and it can enhance its video quality by receiving other layers. Enhancement layers cannot be decodable independent to base layer. For example in figure 2.11, a peer can join only the tree, which delivers the main layer (blue part), and other peers can increase their reception quality by getting more layers from other trees (red or green part).

Differently from CoopNet, the encoding scheme used by Pals to deliver the data makes one of the trees more important than the others. A failure on the main tree will disrupt the video streaming quality, and will not be mitigated by the reception of the only enhancement layers. The use of layered encoding scheme seems to be less appropriate to multiple-trees approach than multiple description encoding scheme, since one of the tree has much more importance than the others. A failure on the main tree will affect the overall P2P network, similarly to the single-tree approach.

However, the MDC encoding schemes needs that receivers get a large amount of the descriptions to decode the video. It has been observed that MDC generates a lot of overhead, which has to be kept as low as possible for scalability reasons [46, 47]. More precisely, in [47], Ross et al. argues that the inefficiency of MDC largely prevents its usage in practical P2P live streaming systems. Even though MDC seems to be more attractive, the layered coding is probably the encoding scheme to use in practice for designing P2P live streaming systems.

2.2 Mesh-based Architectures

The tree-based architectures organize peers into static overlay topologies, where peers have well defined relationships and get the data from one parent (single-tree) or many parents (multiple-trees). The departure of a peer (*churn*) may disrupt the delivery of data for a large number of peers in the single-tree structure. Even though the multiple-trees approach aims to solve this problem, the management of multiple-trees under a highly dynamic environment is still very challenging.

The mesh-based approach has been introduced in recent years with BitTorrent [21], a P2P file-sharing systems successfully deployed on the Internet. As opposed to the tree-based architectures, the mesh-based architectures do not construct any regular overlay topology for delivering the data to users; there is no static topology and the peers establish peering relationships dynamically according to the data availability. Instead of constantly repairing a structure in a highly dynamic P2P environment, the data availability among peers guide the peer relationships. The mesh-based architectures may also be referred as the data-driven approach.

Similar to BitTorrent, the peers form a dynamic random mesh and maintain re-

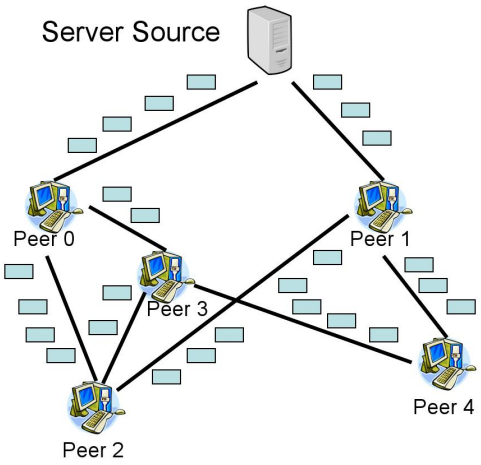


Figure 2.13: Mesh-based architectures: the peers are organized into a transient mesh whose links between peers depend on the availability of data.

lationships with many peers at the same time. They upload/download the content from/to multiple neighbors upon data availability. In the case of the departure of a neighbor, the peer can still get the content from the remaining neighbors. During all their video session, the peers will continue to discover new peers to establish new peering relationships. Since each peer maintains its own neighborhood dynamically, the mesh-based approach is very robust against peer churn. The peers know immediately substitution peers. Overall, some studies have also shown that mesh-based outperform the tree-based architectures [48–51].

In tree-based approach, the video stream is pushed from the source to the peers along the (multiple-)tree structure. In mesh-based, the video parts can be received from different neighbors, invaliding the concept of video stream. As for BitTorrent, the video is divided into small pieces of video, also called *chunks*. Each chunk is spread to all the peers through the mesh. Since chunks can take different paths to reach a peer, they may arrive out of order. A peer will have to buffer the received chunks to put them back into order to play the video.

In the mesh-based approach, one can distinguish two different policies to exchange the chunks between peers. On the one hand, the *push policy*, in which each peer actively pushes a received chunk to its neighbors. A peer could unfortunately push a chunk to a neighbor already having the chunk, wasting its bandwidth resources. On the other hand, the *pull policy*, in which each peer requests explicitly a chunk to its neighbor. Peers periodically exchange data availability information with their neighbors using *buffer maps*. Typically, the buffer map is a vector of bits where the presence of the data is indicated by a bit set to 1, whereas the opposite corresponds to a bit set to 0 [52]. After obtaining the buffer maps from its neighbors, a peer can request

explicitly the chunks and organize its download. The pull policy limits redundant chunk transmissions since a peer get the chunk on request from only one neighbor. However, it adds signaling overhead traffic and additional delay for a chunk retrieval since it needs at least a round trip time (RTT) to request and obtain a chunk.

In this section, we will present several mesh-based architectures for P2P live streaming. At first, we start presenting BitTorrent since it has inspired the mesh-based approach used for P2P live streaming system. Then, we introduce DONet, the first proposed mesh-based P2P live streaming architecture. We further present other architectures that extend or improve the mesh-based architecture by adding new mechanisms such as incentive mechanisms or network coding capabilities.

2.2.1 BitTorrent

BitTorrent [21] is a popular P2P-file sharing system that has been presented in 2003. BitTorrent is a P2P system for the distribution of large file. Even though it has not been designed for the distribution of live streaming, it has inspired the mesh-based approach for P2P live streaming. Indeed, BitTorrent does not build a strict overlay topology for the peers. It lets the peers organize the P2P network into a transient mesh whose links are between peers depending on their interest in available data. The mesh-build is therefore more resilient to the churn of peers since the relation between peers are not clearly defined and are guided by the availability of data. With this transient mesh topology, all the peers can be involved in the transmission and there is no "leaf node" concern as with the tree-based approach.

BitTorrent divides a file into a collection of blocks also called *chunks* (typically 64-512 KB), and peers have to recover all the blocks to download the whole file. The architecture of BitTorrent is illustrated on figure 2.14, and we will explain all the steps for a peer to download a file –peer 5–.

First, peer 5, interested in a file F, must browse dedicated web servers such as "mininova" [53] or "thepiratebay" [54] in order to obtain the corresponding *torrent* file of the desired file F. These web servers are centralized servers dedicated to the storing of millions of torrent files. A torrent file contains information (meta-info) about the file to be downloaded: address of a *tracker* for the file F (URL), name, size, number of chunks of the file, SHA-1 fingerprints of the chunks to check the data integrity. Second, peer 5 contacts the tracker, which coordinates interaction between peers downloading the file and maintains a list of active peers. The tracker sends a subset of the active peers to peer 5 (peer 1 and peer 3 in our example). Third, peer 5 uses peer 3 and peer 5 as a bootstrap in the system and peer 5 connects with them. All the peers downloading the file are part of the *torrent* for the file F, or *swarm*. Among the peers in the torrent, one can distinguish two kind of peers called *leechers* or *seeds*. The *leechers* are the peers that are still downloading the file (peer 1, 2, 3, 5 on figure 2.14). The *seeds* are altruistic peers that have already downloaded the file and stay in the torrent to serve chunks to other (peer 4 on figure 2.14).

To download the chunks, peers exchange with each other a *buffer map*, that is, information about which data blocks they own, and which they want to recover. Thus,

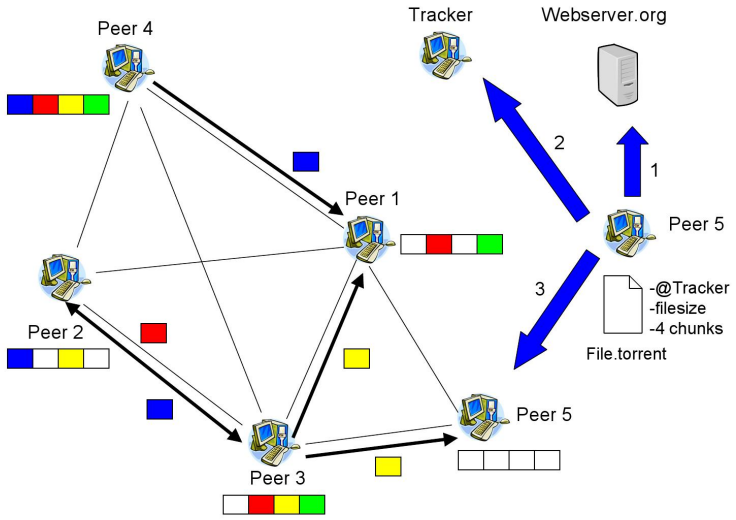


Figure 2.14: Distribution of a file with BitTorrent.

the peers organize the P2P network into a transient mesh whose links are between peers depending on their interest in available blocks. For instance, the buffer map of peer 3 would be 0111, and 1010 for P2 etc. Peer 2 and Peer 3 are interested with each other chunks (red and blue chunks respectively), and peer 1 is interested by the yellow chunks owned by peer 3.

We now focus on the download process for peer 5. The efficiency of BitTorrent relies on its two main algorithms to organize the download of chunks: (i) an incentive mechanism to enforce peers to collaborate and allow fairness between peers in the P2P system; (ii) a scheduling algorithm to select which chunks have to be downloaded first.

The incentive mechanism in BitTorrent rests on reciprocal exchange of data between peers (*tit for tat*). More precisely, a peer transmits a block to another peer if the either transmits one in return. Peers are thus encouraged to contribute to the peer-to-peer system since they are repaid by receiving new blocks. For instance, peer 2 and peer 3 exchange reciprocally chunks of data. A newcomer in the torrent such peer 5 does not possess any chunks to exchange with other. Clearly, it must be allowed to recover its first blocks without having yet transmitted any. To this end, BitTorrent uses an *optimistic unchoke mechanism*, which allows transferring data for free to newly joined peers without any reciprocal exchange. This is the case for peer 3, which unchokes peer 5 and sends to it its first chunk.

The scheduling algorithm in BitTorrent uses a *rarest first* policy to select the chunks to download first. Among its neighborhood, a peer downloads first the chunks with the less number of copies. Peer 5 knows peer 3 and peer 1, which together possess two copies of the red and green chunks and only one copy of the yellow one. Peer 5 starts to

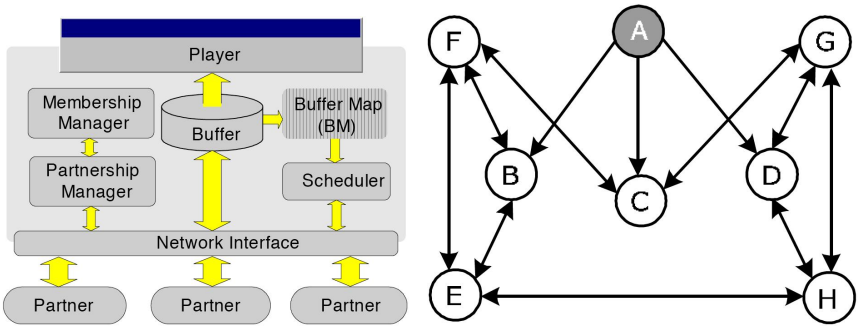


Figure 2.15: The architecture of DONet (left) and the mesh-based topology (right) [12]

download the yellow chunk first. The rarest first policy ensures all blocks stay available in the torrent. When it has finished its download, an altruistic peer can stay in the network to serve other peers and become a *seed*.

Even if BitTorrent proved its efficiency to distribute large files, it can be cheated by strategic clients, or by *free-riders*. The free-riders are peers which download a file without contributing with any chunks. For instance, BitTyrran [55] shows that the tit-for-tat incentive mechanism of BitTorent is not robust to strategic clients, which can improve their download performance for the same level of upload contribution. BitThief [56] exploits the *optimistic unchoke mechanism*, and it is able to download content without contributing with any data.

2.2.2 DONet

DONet [12] is the first mesh-based architecture proposals and takes its inspiration directly from BitTorrent. As explained before, DONet does not build a strict topology to organize peers, but allows peers organizing themselves dynamically upon data availability (data-driven). DONet was also the first architecture to be really implemented and experimented at large scale. Thus, the architecture is well defined and consist of three main modules as shown in the figure 2.15: -"Membership manager", which helps the peers to maintain a partial view of the overlay network; - "Partnership manager", which establishes and maintains partnership with other peers; -"Scheduler", which schedule the order of the data transmissions.

A New joining node contacts the origin server (i.e. video source), which acts as a tracker in BitTorrent. The origin server transmits to it an initial list of random peers already in the video session. Then the new peer knows peers to bootstrap in the network and try to establish peering relationships. During all its video session, each peer discovers new peers by exchanging randomly its list of peers to other peers. To this end, DONet uses a gossip-like membership protocols such as SCAMP [57]. Then, each peer maintains a membership list, which is a partial view of other active peer members in the P2P network.

The video stream is divided into chunks and the availability of the chunks in the buffer of a peer is represented by a buffer map. DONet uses a 120s buffer map, where each chunks counts for 1s video stream. Each peer periodically exchanges its buffer map with its partners. DONet uses a pull policy to receive the chunks: each peer has to request explicitly a chunk to receive it. Based on the information of the buffer maps, and thanks to the scheduler module, a peer will send requests to its partners for the chunks to be fetched and will also be requested to transmit chunks to partners.

As opposed to BitTorrent, which firstly downloads the rarest chunks (rarest-first policy) to ensure the chunks dissemination in the network, DONet does not use the same policy. The live video presents time constraints and there is no need to obtain a chunk after its playback time. The scheduling policy of DONet is to download the chunks with the earlier playback deadline.

The DONet mesh-based architecture is robust to the churn of peers since an affected peer will detect the departure after an idle time, and can quickly react by re-scheduling its chunk request to other partners. DONet also encourages peers to discover new peers and establishes new partnerships with randomly selected nodes. It helps to prevent quickly the effect of the departure of a peer but also to find partners of better quality, which have higher upload capabilities and available chunks.

Since DONet is a mesh-based architecture, one of its drawbacks is the need of a large buffer to download the video chunks and re-order them to obtain a smooth playback quality. This will lead to long start-up delay to play the video and a long lag between peers in the overall networks. Some peers watching the same video can be separated with several minutes in their play-out point.

2.2.3 GridMedia

GridMedia [13] is a mesh-based architecture that aims to overcome the pull policy limitations by introducing a hybrid pull-push policy [58]. The pull policy adds signaling traffic overhead because of the buffer maps exchanged periodically between peers and the specific requests to get the chunks. Moreover, it adds latency since three steps should be accomplished to download a chunk: first, the sender announces the availability of a chunk to the receiver (buffer map); second, the receiver requests the chunk; third, the sender sends the chunk to the receiver. At least three end-to-end delays are involved in these steps, increasing the delay to download a chunk. However, the pull policy is robust with dynamic environment.

For GridMedia, the pull policy is used as a start-up to get the first chunks and after that, each peer will forward a received chunk to its neighbor without any explicit requests from its neighbors (push policy).

The video stream is divided into n sub streams and each sub stream is composed of the chunks whose sequence numbers have the same value modulo n. When a peer joins, it first asks its neighbors to send buffer maps, and then pulls the required chunks according to the packet availability. Once a chunk of a sub stream is requested successfully from a neighbor, the peer will send to its neighbor a sub stream subscription to let it push the rest of the chunks in the same sub stream. In the pull-push policy,

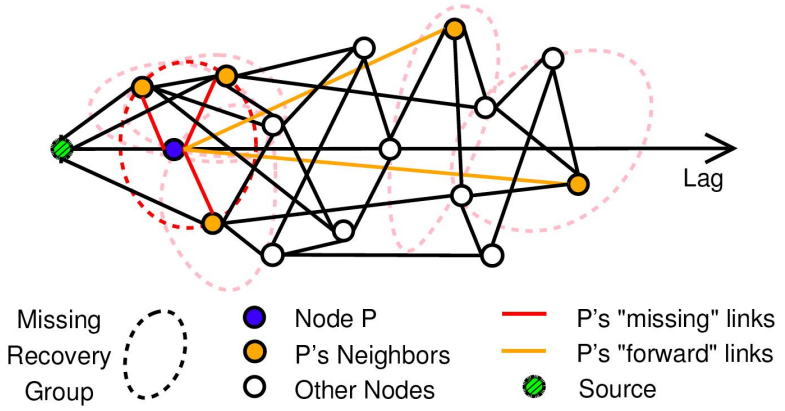


Figure 2.16: The Pulse Architecture [59]

there are pulled chunks and pushed chunks. When over 95% of the chunks are pushed from the neighbors, the peer will stop requesting buffer maps. And once the delivery ratio drops below 95%, the peer will start requesting buffer maps and pull the chunks from its neighbors.

Since a peer subscribes to each sub stream only once, the peer will not receive the same chunk several times; it avoids the waste of bandwidth in the common push policy. The use of the push policy will also reduce the latency to download a chunk. The pull policy is used only as a start-up and the signaling traffic overhead (buffer maps, chunks requests) is limited. At the steady state of the video delivery, the push policy is used to deliver the chunks. In this case, this architecture recalls the multiple-tree approach, where many parents were responsible for pushing the data to child peers.

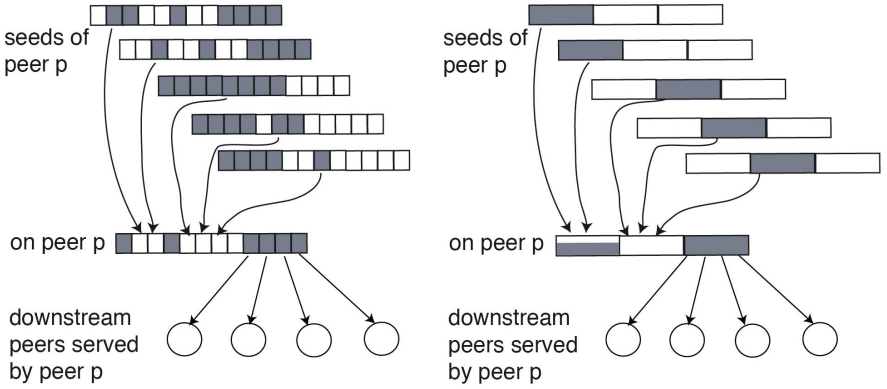
With a highly dynamic environment, the push policy will become impracticable since the sub streams pushed to a peer could be frequently broken. Then, the new chunks will have to be continuously requested and pulled by peers, similar to the classical pull policy used with DONet, which limits the benefits of this hybrid policy.

2.2.4 Pulse

Pulse [59] is a mesh-based architecture similar to DONet using gossip membership protocol and pull policy to exchange the data. However, in Pulse, the peer associations are driven by incentive mechanisms based on implicit feedback information, such as the amount of data exchanged in the recent past and the current reception delay of the stream.

Some recent studies [60] analyze the effects of the tit-for-tat incentive on BitTorrent. They show that it allows the best resource contributors to associate together and get higher download performance. The "clustering effect" between peers with the same amount of resources has in turn a positive effect on the global system performance.

The Pulse incentives combine together two mechanisms: a primary optimistic tit-for-tat peer selection policy, and an additional excess-based altruistic incentive. In-



(a) Traditional P2P live streaming architecture

(b) Architecture of R^2

Figure 2.17: Differences between traditional P2P live streaming architecture (left) and R^2 (right) [61].

tuitively, the primary mechanism should foster cooperation among resourceful nodes, while the other should both facilitate peer discovery and allow the richest nodes to contribute more effectively to the system.

These two incentive mechanisms are presented in figure 2.16. The peer P is trading data with two groups of peers: MISSING and FORWARD groups. Peers in the MISSING group are selected to favor reciprocal data exchanges, and in a way to insure that both parties have chunks the other one needs (optimistic tit-for-tat). The FORWARD group helps to introduce a component of altruism in the system and, at the same time, allows resourceful nodes to contribute more of their upload capacity to the system (excess-based altruistic incentive).

In the Pulse architecture, the fundamental role of incentives is not to enforce fairness between peers (i.e. equality of contribution). Instead, the use of incentives as a local policy to exchange data between peers helps to optimize the system for better global performance.

2.2.5 R^2

R^2 [61] proposes to take advantages of random network coding [62, 63] to deliver the video stream efficiently. In their previous work, the authors proposed Lava [64], a mesh-based architectures similar to DONet, with network coding capabilities. The network coding module was implemented as a "plug-in" component to a classical P2P live streaming architecture. While a traditional P2P streaming protocol sends original segments (*chunks*), the random network coding makes it possible to send coded blocks of each segment, such that receiving peers may decode them on-the fly. The authors claims that with Lava, network coding provides some marginal benefits when peers are volatile with their arrivals and departures, and when the overall bandwidth supply

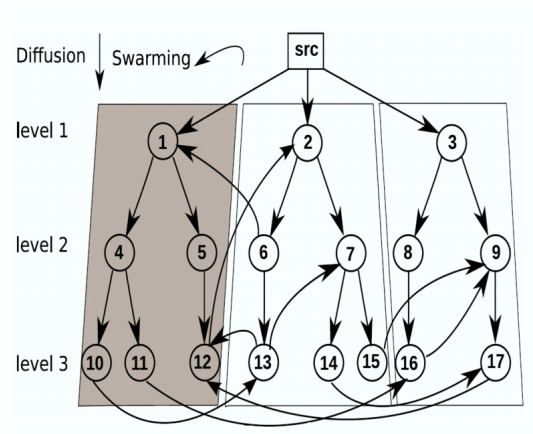


Figure 2.18: The PRIME architecture [65].

barely exceeds the demand. However, the authors are still convinced that random network coding is beneficial for P2P Live streaming. They design therefore R^2 to incorporate random network coding with randomized push policy.

In R^2, the peers are still organized into a mesh and exchange buffer maps between each other. As opposed to Lava, the receiver peers does not request explicitly a segment (pull policy). A sender peer will randomly push a block of segment among all the remaining segments which are not completely downloaded by the receiver peer. The feedback from the receiver peer is essential. In contrast with traditional mesh-based architecture, the receiver peer sends its buffer map whenever its buffer status changes: that is to say when it has completed the downloading of a segment. The extensive transmission of the buffer maps could generate an excessive traffic overhead. R^2, instead, divides the buffer into 40 segments (120 for DONet) of 4 seconds each (1 second for DONet) to keep a low signaling traffic overhead. In traditional mesh-based architecture, a missing segment is downloaded from a single peer. In R^2, with random push coupled with random network coding, a segment can be downloaded from multiple peers. This is due to the fundamental characteristic of the random network coding, in which any coded block is as good as any other, regardless of the peer that produces them. The contrast between traditional architecture and R2 is shown in the figure 2.17.

2.2.6 PRIME

In PRIME [65], participating peers form a randomly connected and directed mesh. Each participant peer has multiple parents and multiple child peers. Each peer tries to maintain a sufficient number of parents that can collectively fill its incoming access link bandwidth. When a peer needs new parents, it contacts a bootstrapping node to learn about a random subset of other participating peers in the system and then requests those peers to serve as its parents.

The content delivery mechanism in PRIME combines push reporting by parents with pull requesting by child peers. The content is encoded with Multiple Description

Coding (MDC) which enables each peer to maximize its delivered quality by pulling a proper number of descriptions. Each peer, as a parent, progressively reports its newly received packets to all of its child peers and as a child, periodically requests an ordered list of packets from each one of its parents.

The video content is divided into a set of segments, and each segment is composed of data units. The different data units of the segment should be rapidly delivered (or diffused) to different subset of peers. Then, participating peers can exchange (or swarm) their data units and contribute their outgoing bandwidth until each peer has a proper number of data units for the segment. In PRIME, the delivery of a segment follows a two-phase approach: a diffusion phase and a swarming phase. The global pattern of the content delivery is illustrated on figure 2.18

Once a new segment becomes available at the source, it serves different data units to its child peers, which in turns push the data to their descendant peers. Each peer p as well as all of its descendant peers in a so called "diffusion sub-tree" rooted in p receive the same data unit of each segment during the diffusion phase of that segment. In figure 2.18, one of the three diffusion sub-trees that is rooted at peer 1 is shaded. At the end of the diffusion phase of a segment, all peers in the overlay have at least one data unit of that segment.

At the beginning of the swarming phase of a segment, participating peers can only pull the missing data units of the segment from their parents that are located in different sub-trees. For example, in the figure 2.18, peer 9 can obtain a new data unit from peer 15 but not from peer 16. The swarming connections are represented with curly arrows. in figure 2.18. If a peer knows enough swarming parents located at different diffusion sub-trees, the peer can pull the missing data units from all parents in a single swarming interval (peer 12 in figure 2.18. Otherwise, the peer experiences a content bottleneck (e.g. peer 9 in figure 2.18) and thus requires more than one swarming interval to obtain the remaining data units. During these extra intervals, some of its swarming parents will obtain new data units of the target segment, and can pass them along. For example, peer 16 receives a new data unit from p11 after one interval and can pass it to peer 9 in the next interval.

2.2.7 Chunkyspread

With Chunkyspread [66], the philosophy of the authors differs from the general case. To their meaning, whatever the multiple-tree or mesh approaches, the content is always spread into a tree, and the difference is whether each of the packets traverse its own tree (per-packet or mesh-based approach), or a continuous flow of packet traverses always the same tree (per-slice or multiple-tree based approach). In this last case, there will be as much trees as slices (i.e. multiple-trees). With such definitions, they defend a new and original point of view for the live streaming delivery in P2P overlay network.

The mesh-based approach could generate a lot of signaling traffic overhead because of the pulling of packets. It also requires that packets be buffered long enough to accommodate the delay, avoid packet loss of the swarming strategy and get a smooth playback quality. With the multiple-tree approach, in an environment with constant

churn, trees are continuously being destroyed and rebuilt. It results in a considerable control message overhead. In order to avoid packet loss due to disruptions in the trees, nodes must also buffer packets for the period of time it takes to repair a tree. At the end, both approaches exhibit the same types of trade-offs: control message overheads, suffer from substantial delays in packet reception and require some amount of buffering to prevent packet loss (though for different reasons).

For the authors, the right approach to choose should be the simplest one. The building and repairing of trees are not simple tasks but pushing the data to receivers is easy to achieve. The mesh-based approach with the swarming strategy seems simpler to them rather than building trees. Nevertheless, the mesh-based approach must minimize the overhead and delay of the push-pull policies, and it is very challenging with random mesh structure. Finally, they argue that there is some complexity to build any kind of approaches but not as the same level: the complexity is for building the tree structures (multiple-tree) or delivering the data (mesh-based).

Based on their previous work SwapLink [67], the authors prefer multiple-tree rather than mesh-based approach. They propose Chunkyspread, a multiple-tree architecture with which they argue that tree-building can be made as simple as mesh-building. The foundation of Chunkyspread is the ability to build random sparse overlay graphs with tight statistical control over heterogeneous node degrees.

Like Splitstream, the source transmits the stream as M distinct slices to each of these neighbors. These neighbors become the roots of the multiple-trees. But, quite unlike Splitstream, the trees are not necessarily node-disjoint. To join in the Chunkyspread architecture, peer must first contact a rendezvous node at a well-known location This rendezvous node must know of at least one existing member. Once a joining member node finds at least one existing node, it participates in a continuously running distributed algorithm called Swaplinks that produces a random graph among all nodes using simple weighted random walks. This random neighbor graph is the underpinning of Chunkyspread.

Swaplinks is able to statistically control the node degree of each node, and Chunkyspread exploits this to give nodes with higher target loads proportionally higher node degrees. The idea here is that nodes with higher load should have more neighbors to transmit slices, and nodes with lower load should have proportionally fewer neighbors. A node, upon joining the random graph, tries to find a parent for each slice without forming a loop. The loop are detected and avoided by using bloom filters [68] in the data packets.

Regarding our classification, we chose to place Chunkyspread into the mesh-based approach because the peers are organized with a random subset of other peers, as with a random mesh, and then they exchange the data with their neighbors. The notion of data delivery tree come from the fact that peers are synchronized together with their play-out time. Even with buffered data, a peer receiving data is not able to reciprocate because it is further in the play-out time: it has no data of interest for its provider peer. It can only retransmit data to other peers above in the play-out time. The video live streaming, by its continuous nature and temporal constraints, impose the data to be spread into an oriented path such a tree. This phenomenon has been observed in our measurement studies and will be discussed further in this thesis. Chunkyspread

Table 2.1: Architectures Summary

Name	Control Plane	Observations
Single Tree		
PeerCast	tree	inspired from IP-multicast
Nice/ZigZag	clusters	ZigZag balances the load of the cluster leaders
Narada	mesh	small-scale network
Bullet	mesh/tree	first step to multiple-tree
Multiple-Tree		
Splitstream	DHT	disjoint tree for each slice of data
CoopNet	centralized server	Multiple Description Coding
Pals	-	Layered Coding
Mesh-based		
DONet	gossip algorithm	pull policy
GridMedia	gossip algorithm	hybrid push-pull policy
Pulse	gossip algorithm	incentive mechanisms
R^2	gossip algorithm	network coding and push policy
Prime	centralized	push/pull policy; MDC
Chunkyspread	random walk	tree per-packet vs. tree per-slice

illustrates that it is difficult to associate each architecture into a strict category. In fact, it depends whether we refer to the organization of peers (control plane) or to the data delivery (data plane).

2.3 Summary

In this chapter, we presented the major P2P live streaming architectures. We classified these architectures into two main categories: tree-based and mesh-based. The tree-based category is also divided into the single-tree and the multiple-tree approaches. This survey is summarized on table 2.1.

The single-tree approach comes first with the application-layer multicast, and it takes its inspiration directly from the IP-multicast by building a delivery tree among all the receivers. Since the single-tree approach suffers drastically from the churn of peers or the under-utilized resources of leaf nodes, the multiple-tree approach has been introduced to cope with these limitations. In addition, the peers can then adapt the quality they want to receive and also contribute with their resources, even if they have heterogeneous capabilities and are not able to provide the full data. Then, with the recent success of BitTorrent, the mesh-based approach appeared and is currently admitted as a standard reference for P2P live streaming. By authorizing many providers of data, the mesh-based approach enables every peers to contribute with their resources, without building a strict topology between peers as with the tree-based approach. It

is also more resilient to churn of peers since peers receive the data from several peers at the same time.

2.4 Discussion

DONet is the first example of such mesh-based approach and most of the other presented mesh-based architectures slightly improve or refine it by adding functionalities such as incentive mechanisms, hybrid push-pull policy or network coding capabilities.

Only DONet (i.e. Coolstreaming) and GridMedia have been recently implemented and deployed on the Internet. Both announce peaks from 80,000 to 220,000 concurrent users at the same time [58, 69–71], demonstrating roughly the feasibility of such architectures.

To put it into perspectives, TF1, the most popular TV broadcaster in France, has an audience that ranges from 5 millions to 10 millions of concurrent viewers [72]. In France, for some events such as soccer games, the peak of audience could reach about 20 millions of concurrent spectators: a third of the overall French population. Some worldwide events such as the soccer World Cup, Olympic Games or Superbowl can reach hundreds of millions of spectators at the same time. Of course, these numbers are the cumulative audience for all the TV broadcasters and not the audience for a single one.

However, the P2P applications are used worldwide and could target such very large audience. A scenario could be the Olympic games, where a source broadcast in live the athletics 100 m final to millions of spectators located all over the world. Thus, the deployments and experimentations of the presented P2P Live streaming architectures are still very limited and it is still unclear if these architectures could scale to very large audience (one or two order of magnitude larger).

In parallel, many P2P-TV applications have been released on the Internet and claim to take their inspiration from BitTorrent and the mesh-based approach. These applications counts already millions of downloads and registered users [73]. The most popular one is probably PPLive [14] but many of them are getting popularity such as PPSTream, SOPCast, TVAnts, UUSee, TVUPlayer, PPMate ([15–20]). These applications are proprietary and their source codes are not available: there is no implementation or architecture details about these commercial applications. The exact way of working of these applications is totally unknown, but they are expected to be massively used in a near future. It is therefore important to study these commercial P2P-TV applications in order to get knowledge about them.

In this thesis, we will investigate deeply the commercial P2P-TV applications. This study will allow us to uncover the interesting properties of these P2P-TV applications and the characteristics of their users. This study could also help for the future design of new mechanisms or paradigms for P2P live streaming.

Chapter 3

Peer-Centric Experiment of P2P-TV

The peer-to-peer has become the architecture of reference to allow a television service on the Internet. The main reasons are its reduced infrastructure costs and its self-scalable properties. There is no need to deploy any new infrastructures (as for CDN servers) or to modify the current infrastructure of the Internet (as for IP-multicast), which would be very costly to achieve. With P2P, the users act as both client and/or server: they download the content, and are also responsible for forwarding it to other users. The users contribute with their resources in the network. The amount of resources increases with the number of users. The video is not only provided by a single server, and the inherent bottleneck problem of the client/sever model is overcome. Thus, P2P is able to scale to a wide audience, which is expected for a television service.

As explained in the previous chapter, we noticed that many P2P architectures have already been proposed to stream live multimedia flows. Only a few of them have been truly implemented or experimented [12, 13], but at a limited scale. Most of these architectures proposals rely on simulations to validate the proposed architecture and mechanisms. In parallel, many commercial applications have been released on the Internet [14–20] without providing any architecture or implementation details. There was no knowledge about these systems and no information was available to evaluate them properly.

Thus, in order to fill the gap between architecture proposals and commercial applications, we performed an extensive traffic measurement experiment of commercial P2P-TV applications and analyzed the traffic. Our main goal was to obtain a better understanding of these systems.

Moreover, the popularity of P2P-TV applications is increasing. Since the P2P traffic and the video traffic constitute already the most important part of the Internet traffic, it is therefore expected that P2P-TV will contribute to increase the traffic of the

Internet and count for a significant amount of this traffic. Hence it is also important to evaluate the impact on this traffic on the network.

In this chapter, we present our P2P-TV measurement experiment. We first introduce the experimental testbed and then our analysis of P2P-TV traffic.

3.1 Description of the Experiment

With the aim to better understand traffic properties, users' behavior and underlying mechanisms of P2P-TV, we considered four applications during a worldwide event. Analyzing different applications allows studying such systems without being too closely related to the design of the applications and thus making the results more general.

We collected traffic traces during the 2006 FIFA World Cup from June 09 to July 09 because we believe that it can be representative of events of interest in P2P-TV communities. The 2006 FIFA World Cup represents indeed one of the biggest worldwide sport events that attracted tens of millions of viewers from all over the world. The mobile network operator "3" reported that the 2006 FIFA World Cup pushed usage of their implementation of mobile TV to an all time high with over 3.6 million viewings of its World Cup-based mobile programming [74]. Some of the strongest motivations for people to resort to P2P-TV to follow the soccer matches were that (i) in several countries the matches were happening during working hours, when people had only PCs available, and (ii) not all matches were broadcast by some national TVs or were broadcast only by Pay-TV systems. Moreover, users did not need to understand the language on the audio channel to enjoy the video, making possible the creation of a content distribution network ranging worldwide.

In the next subsections we give some background information on the applications analyzed and details on the measurement setup.

3.1.1 Studied P2P-TV Applications

These applications are mainly used to watch TV programs that are not available for users in their own country. This is because of copyright policies but the legality of the delivered content with P2P-TV applications is out of the scope of our study. As an example, some sports events could not be available with local TV broadcasters but can still be watched by using these applications. In this case, a source, situated in a country where the TV event is available, delivers the content with the P2P application, and each user becomes a peer that downloads and forwards the content to other. An user willing to watch a special TV program with P2P-TV browse dedicated website [75] reporting the schedule of the P2P-TV channels, among many different kind of applications and programs. The user will have to browse the channels of the application or to click on the channel URL provided by the website to start watching the TV.

For our experiments, we chose the applications PPLive, PPStream, SOPCast and TVAnts, because they are among the most popular. Actually their users, on the community website at [75], ranked these applications among the best and efficient applications

to watch live television. Nowadays, these four applications are still very popular and, e.g. in the case PPLive [76], count millions of concurrent users.

All the largely deployed P2P-TV systems claim to use a mesh-based architecture as those presented in chapter 2. The mesh-based architecture for P2P-TV systems takes its inspiration from BitTorrent [21] and use the same kind of swarming protocol, as in DONet [12]. Instead of building a strict topology (e.g. a broadcast tree), a mesh is built among peers whose links (peering relationship) depend on the data availability on each peer. The topology is dynamic and will continuously evolve according to the peering relationship established between peers. As we explained before, with the mesh-based architecture, the video flows are divided into data chunks and each peer downloads the chunks from other peers concurrently. To get knowledge of the available data among peers, the peers exchange information with each other representing the data they have *buffer map*.

Thus, in these P2P protocols, there are two kinds of traffic: video traffic which is used for exchanging data chunks, and signaling traffic used for exchanging the information needed to get the data. Thanks to the signaling, the peers know how the video data chunks can be downloaded by exchanging randomly with other peers information about the data chunks they own (buffer map) and the neighboring peers they know. Therefore, with such signaling traffic, each peer discovers iteratively new peers and new available data chunks.

However, even if these applications are freely available and affirm to use a mesh-based architecture, their source code is not open and their exact implementation details and protocols used are still widely unknown. Therefore, we can only count on traffic analysis to understand their transmission mechanisms and peers' behavior.

3.1.2 Measurement Experiments Testbed

We collected a huge amount of data, measuring most of the World Cup soccer games with four different applications at the same time. We captured the entire packets with their payload. The total size of our traces collection is about 250 GB.

In this work, we focus on four packet traces, one for each application, collected on June 30 2006 in the campus network of the Université Pierre et Marie Curie - Paris 6. From our collection, we selected these traces because on that day two very important quarter-final matches were played, which attracted a lot of P2P-TV users. The traces are publicly available at [77]. It is worth stating that we also analyzed the other collected traces and we obtained results similar to those here-presented.

On the selected day, two quarter-final matches were scheduled: *Germany vs. Argentine* in the afternoon and *Italy vs. Ukraine* in the evening. The choice of this day was motivated by non-technical issues too: in order to have the highest number of users involved in the trace we collected, we considered matches with favorite teams, team of the hosting country, etc. During each match, we used two computers, each one running a distinct P2P-TV application as well as `WinDump` [78] to collect the traffic. Therefore we collected two traffic traces for each match, one for each application. In particular,

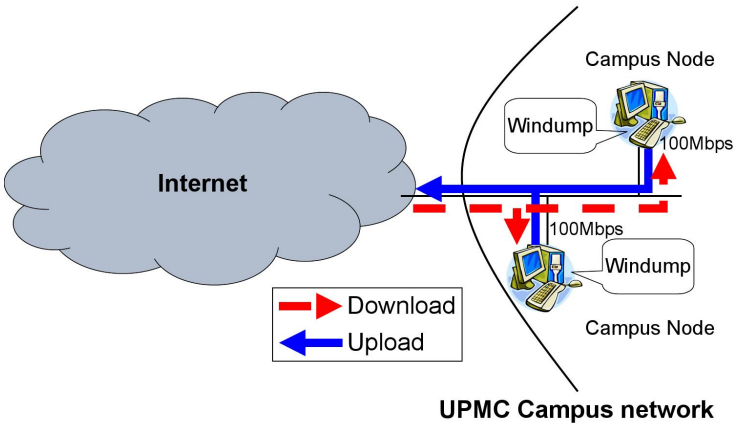


Figure 3.1: Measurement experiments testbed. Each node is a common PC directly connected to the Internet via the UPMC campus network.

we respectively collected traffic from PPStream and SOPCast during the first match and from PPLive and TVAnts during the second one.

Our measurement testbed is described in figure 4.1. To collect packets, we used two PCs equipped with 1.8GHz CPUs, common graphic card capabilities, and running `Windows XP`. The PCs were situated in the campus network and were directly connected to the Internet through a 100Mbps `Ethernet` link.

For all the measurement experiments, the consumed bandwidth was always relatively low and did not exceed 10Mbps. The Ethernet cards did not suffer any packet loss and captured all the packets. For all the experiments, the nodes were watching *CCTV5*, a Chinese TV channel available for all the measured applications. It was important to watch the same TV channel with all the applications to be sure that the peers' behavior was similar in each trace. For example, despite the different applications, during the advertisements a user may stop watching the channel switching the application off and then switching it on a few minutes later. All the applications used an MPEG4 codec, which mixes video and audio content.

After collection, the traces had to be cleaned by removing packets not related to the applications. This operation is necessary because we do not know the characteristics of the traffic of such applications. Therefore, we first captured all the traffic exchanged by the nodes under test. After that, we inspected the traces and filtered out traffic not related to the observed applications such as *NTP* or *ARP* packets, which are commonly used into networks. This was done offline and both manually using `tcpdump` with specific filters and using Plab [79], a software for traffic analysis at packet-level that we also used to obtain packet-level, flow-level, and host-level presented in this chapter.

Table 3.1: Summary of packet traces.

	PPLive	**PPStream**	**SOPCast**	**TVAnts**
Duration (s)	13,321	12,375	12,198	13,358
Size [MB]	6,339	4,121	5,475	3,992
Download[%]	14.11	20.50	16.13	24.76
TCP	14.09	20.50	0.23	14.71
UDP	0.02	≈ 0.00	15.90	10.05
Upload[%]	85.89	79.50	83.87	75.24
TCP	85.81	79.50	3.89	61.67
UDP	0.08	≈ 0.00	79.98	13.57

3.2 Understanding P2P-TV Traffic

In this section we analyze traffic characteristics in detail. In particular, we first describe some general properties of this traffic, then we discuss issues related to the separation of video and signaling flows, and we show distinct results for them. Finally we present an analysis of the time-scaling behavior because it has been shown in the literature that this is an important property of network traffic that can impact on performance of network nodes [80].

The considered applications generate traffic using different ports and protocols. Table 3.1 contains the information regarding the studied traces. The time duration of the collection ($\approx$ 225 minutes) is longer than that of a soccer match ($\approx$ 105 minutes). We chose to collect the traffic before and after the games to capture all the effects that the live interest on a soccer game could produce on the peers' behavior (e.g. flash crowds).

We observe that there is much more traffic in the upload direction (i.e. from our controlled node to the other peers, blue solid line in figure 4.1) than in the download one (i.e. from all the other peers to our node, red dashed line in figure 4.1). As an example, figure 3.2 shows the total download and upload throughput for TVAnts. The download throughput is quiet constant and slightly above the video bitrate while upload fluctuates largely and at a higher rate. The results for all the other applications are similar and not presented here but can be found in [81]. Our nodes attempt to download the video at a constant bitrate and have wide upload capacities. This is due to the fact that our computers are connected to the Internet through a 100 Mbps Ethernet link. Therefore, in contrast with more common ADSL connections, we have equal upload and download capacity. This implies that, as shown in the following section, we are able to provide video chunks to a large number of peers.

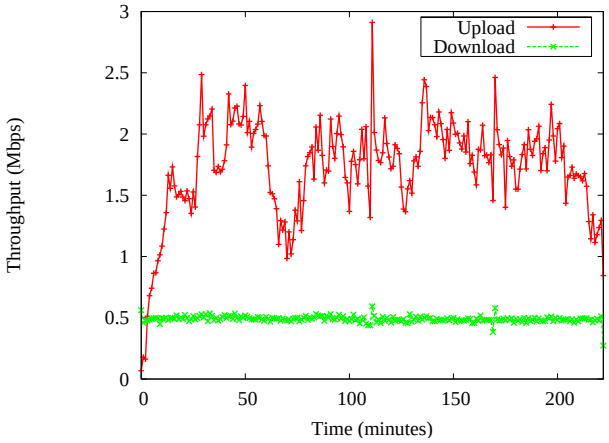


Figure 3.2: Example of total download and upload throughput for TVAnts. All the applications have the same throughput pattern.

3.2.1 Transport Protocols

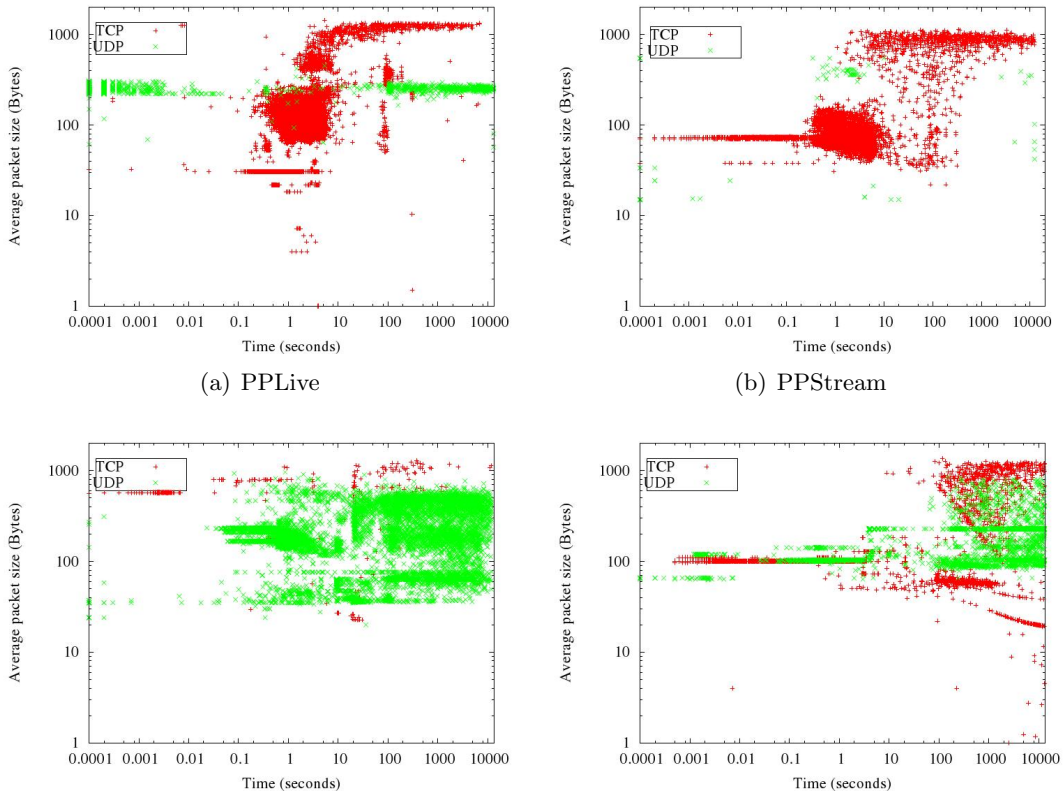
As stated before, these applications use signaling traffic to exchange information between peers and video traffic to exchange video chunks. In this way, a peer is able to add extra signaling and video sessions with other peers.

The video sessions are likely to have long duration: when a peer discovers another peer to exchange data with, it is better for it to keep alive these sessions as long as possible and beneficiate of the data exchange. The signaling sessions are likely to be shorter in time because they only transport information to let peers decide whether to establish video sessions or not. Furthermore, the video packets are expected to be large-size and the signaling packets are supposed to be of smaller-size.

The figure 3.3 plots the average packet size according to the session duration using a log-log scale. The "red points" are used for the TCP sessions while the "green points" are for the UDP sessions. Each point represents a single session and its coordinates have the following meaning: the x-axis is for the duration of the session; the y-axis is the average packet size of the session.

PPLive (figure 3.3(a)) and PPStream (figure 3.3(b)) have similar TCP traffic patterns and they use slightly UDP. Their UDP sessions for PPLive vary from short ($< 1s$) to long duration ($> 1000s$) but they are quite constant for PPStream (a few seconds), but their average packet size is quite small and constant ($\approx$ 250 Bytes). The UDP traffic for PPLive and PPStream transports obviously signaling sessions. PPLive and PPStream exhibit two clusters of points in their TCP traffic patterns: the one in the middle of the plot is composed by sessions with short-time duration and short-size packets and is for signaling sessions. The other cluster on the right top of the plot is for video sessions because the sessions have long-time duration and large-size packet. PPLive and PPStream use TCP to transport both video and signaling traffic.

Regarding SOPCast traffic pattern (figure 3.3(c)), we notice that it uses almost ex-



(a) PPLive (b) PPStream

(c) SOPCast (d) TVAnts

Figure 3.3: Average packet size according to the session duration

clusively UDP. We can still observe clusters in the middle (signaling traffic) and on the right top of the plot (video traffic) but they are not so clearly formed. SOPCast transports both signaling and video traffic on UDP. Compared to the other measured applications, TVAnts traffic pattern shows a balanced use of TCP and UDP (figure 3.3(d)). We can distinguish signaling and video clusters but they both contain TCP and UDP traffic. TVAnts transports signaling and video sessions both on TCP or UDP. However, most part of TVAnts traffic is transported by TCP ($\approx$ 75%, table 3.1).

Table 3.2 summarizes our observations from figure 3.3: all the measured applications have different traffic patterns. PPLive and PPStream use only TCP for video and signaling traffic and add a few UDP for some signaling traffic. TVAnts uses both TCP and UDP for all kind of traffic and SOPCast uses almost entirely UDP.

PPStream and PPLive use TCP as a default choice to transport the multimedia content. Differently, SOPCast adopts UDP to transport the video traffic. The presence of a few TCP traffic reveals that SOPCast is also able to switch to another transport protocol when necessary to reach remote end-hosts on the Internet. Besides, in another similar experience, we modified the rules of our firewall to discard all the UDP packets

Table 3.2: Observations Summary for Traffic Patterns.

	PPLive	**PPStream**	**SOPCast**	**TVAnts**
Video traffic	TCP	TCP	UDP a few TCP	TCP UDP
Signaling traffic	TCP a few UDP	TCP	UDP a few TCP	TCP UDP

and we observed that SOPCast was able to rely only on TCP for both signaling and video traffic. It corroborates the fact that SOPCast uses UDP as a default choice but could also use TCP to reach any kind of host on the Internet. Depending on the reachability of the remote end-hosts, TVAnts is also able to use both transport protocols.

If we consider the overall traffic of all the applications, we observe that the video traffic is mostly transported by TCP, which is not the transport protocol of choice for multimedia and real-time applications.

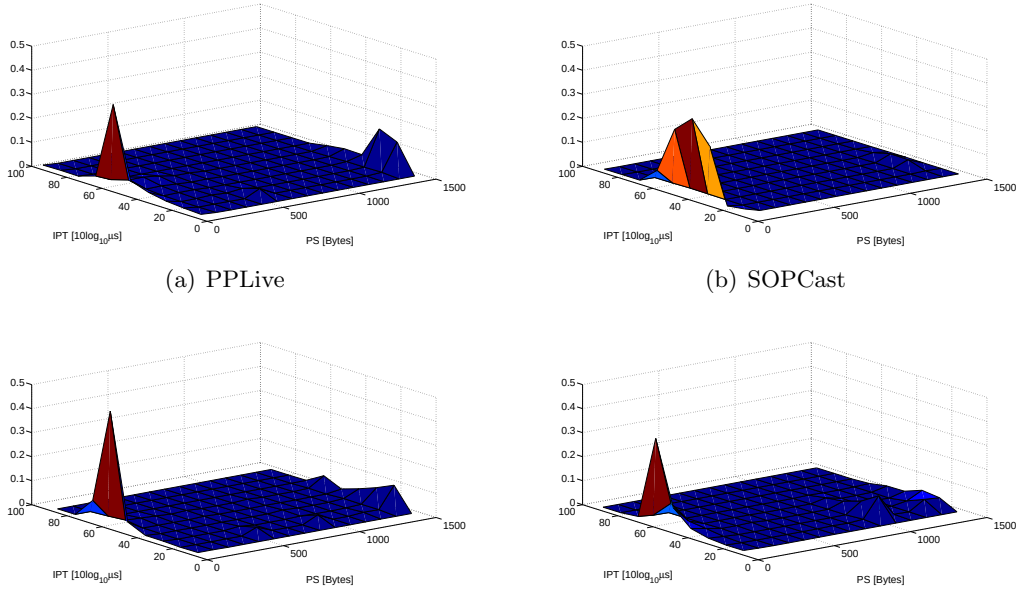
As for every networking applications, TCP comes with its advantages such as reliability but also its disadvantages such as high traffic or processing overhead. It is well know that a TCP connection generates many acknowledge packets that do not transport any useful information but consume bandwidth resources, or use larger header size (min. 20 Bytes) compared with UDP (8 Bytes). Every TCP connections also need to keep states at each end-points, which consume processing resources on each hosts. The reliability of TCP could also become a drawback if a packet retransmission takes too much time to recover from a loss. In this case, the retransmission consumes bandwidth resources and is useless if received after its playback deadline; the user will experiment the packet loss as well. The congestion control of TCP may also limit the bandwidth to download or upload the video chunks. If the achieve download rate at a peer is inferior to the video bitrate, the user experiments a bad quality reception for the video. We suppose the choice of TCP has been decided by the P2P-TV designers to allow the applications to reach all kind of users on the Internet, even though they are behind filtering or Network Address Translator systems [82] (NAT), or set top boxes.

3.2.2 Ports

Table 3.3 shows the ports used by the applications. PPLive and SOPCast present a similar behavior. Indeed, with these applications, the machine under test uses mostly the same ports for all the communications with the other peers which, in turn, use a wide range of different ports. PPStream behaves similarly, except that it uses a fixed remote port and three different local ports for the very few UDP packets. It is also interesting to note that both PPStream and PPLive use the local UDP port 5747. Finally, a peculiar behavior is noticed for TVAnts, which uses port 16800, both local and remote, for most of the UDP and TCP packets. This is probably because TVAnts

Table 3.3: Utilized Port number (percentage of packets).

		PPLive	**PPStream**	**SOPCast**	**TVAnts**
Remote Peers	TCP	Several	Several	Several	16800 (> 25%)
	UDP	Several	7201 (100%)	Several	16800 (> 60%)
Controlled Peers	TCP	10549 (> 99%)	11430 (> 99%)	8516 (> 99%)	16800 (> 71%)
	UDP	5747 (100%)	5747 (42%), 11430 (54%), 65535 (4%)	8516 (> 99%)	16800 (> 99%)



(a) PPLive (b) SOPCast

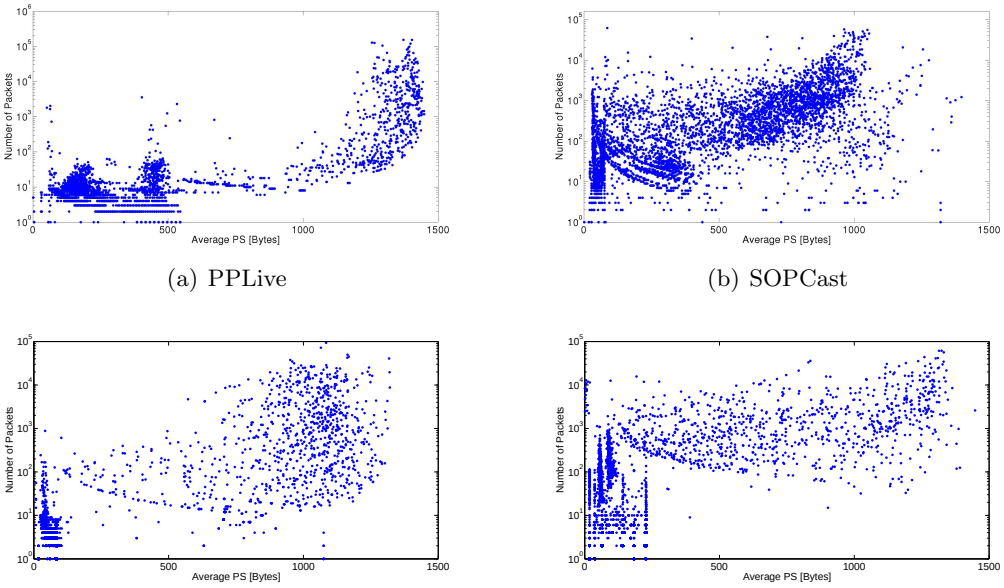
(c) PPStream (d) TVAnts

Figure 3.4: Joint probability distribution of Inter Packet Time and Packet Size.

sets a default port on a new installation that can be changed thereafter by the user. Looking at Table 3.3, it is also evident how P2P-TV traffic cannot be reliably identified by looking at transport protocol ports, motivating the need to find different ways to recognize such application traffic.

3.3 Signaling and Video Traffic

As we explained in section 3.1.1, the P2P applications we studied generate two kinds of traffic: video and signaling. The signaling traffic of P2P-TV systems is not expected to be delay-sensitive, because it is used for exchanging information about peers or data



(a) PPLive (b) SOPCast

(c) PPStream (d) TVAnts

Figure 3.5: Upstream flows: Average Packet Size vs. Number of Packets.

availability but not for interactive commands, as for Video on-Demand (VoD) systems like Joost [83]. In video on-demand systems, the users may want to move the video playback instant forward or backward promptly. In the case of P2P-TV, it is not possible to have this kind of interactive commands since the data flows are broadcasted live. In general we can say that the signaling and the video traffic have not the same characteristics such as packet size or delay constraints, and they would have a different impact on the network. Therefore we want to separate video and signaling traffic in order to analyze their peculiar properties.

Because the protocols adopted by such applications are not open, we rely on a heuristic based on traffic properties. A simple heuristic to separate these two kinds of sessions in PPLive traffic was previously proposed by Hei [84]. Such heuristic works as follows: for each session (same IP addresses and ports), we count the number of packets larger than or equal to 1200 Bytes.

If a session has at least 10 of such large packets, then it is labeled as a video session. All the non-video sessions are supposed to carry signaling information. To understand if it was reasonable to apply such heuristic to all of them, we investigated traffic properties for all of the four applications, driven by the following considerations. It is expected that video sessions are essentially composed of large-sized packets sent at small and regular time intervals, whereas signaling information should be carried by smaller packets sent much less often compared to video chunks. For the same reasons

we expect to find that signaling sessions exchange much less packets than video sessions in general.

Figures 3.4 and 3.5 reveal interesting properties of overall P2P-TV traffic generated by the four considered applications. Moreover, they confirm the above intuitions by showing that there are packets and sessions with different properties and that the packet size property may be a good heuristic to discriminate between signaling and video sessions.

Figure 3.4 shows the joint Probability Density Function (PDF) of the Inter Packet Time (IPT) and Packet Size (PS) of the download traffic. The IPT of each packet is the time elapsed between that packet and the previous one of the same session, and as usual for the PS, we considered the protocol-layer payload size, discarding all TCP packets without payload. For each application, we only considered packets related to the prevalent transport protocol, e.g. TCP for PPLive and UDP for SOPCast. The distributions of these applications are different but, for all of them, we can distinguish two main clusters of packets: small-size packets ($<$ 200 Bytes) with large IPT and large-size packets ($>$ 1000 Bytes) with small IPT. Most of the video packets should then belong to the large PS and small IPT cluster. The signaling packets, instead, should mostly belong to the other cluster with small PS and large IPT.

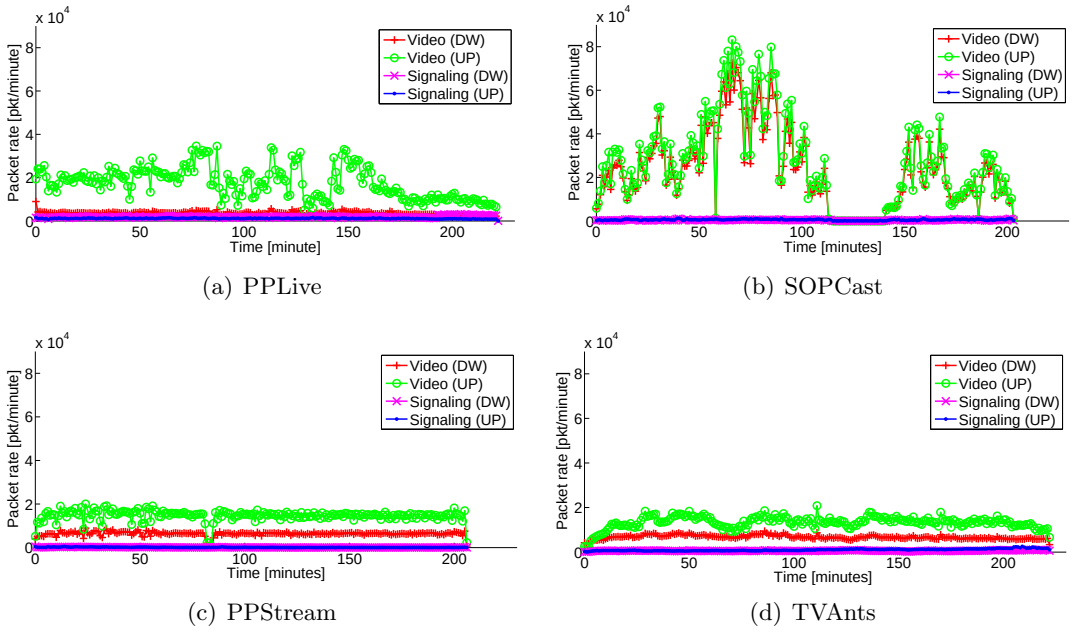
In Figure 3.5 instead, we show scatter plots in which, each point coordinates are given by the average PS and the number of transmitted packets of each session. This last coordinate is plotted on a logarithmic scale axis. In these diagrams, we can see that the sessions with the largest numbers of packets (supposedly video sessions) tend to have high average packet size. Both these results made us very confident that the cited heuristic could be used for all the considered P2P-TV applications. Furthermore, to be sure that this heuristic does not introduce large errors in our analysis, we also manually inspected the traces. This verification allowed us to discover that there are different kinds of signaling packets, that such packets have fixed sizes, and that these sizes are always smaller than 1000 Bytes. Thus, considering also the findings about the PS distributions of the four applications, we modified the heuristic to use a limit of 1000 Bytes instead of 1200 Bytes. Therefore, we can state that, with regard to the traces we consider, the heuristic is effective to discriminate between signaling and video traffic, and we used it to perform separate analysis of them, as shown in the following sessions.

In Table 3.4 we report statistics on the ratio of signaling traffic with respect to overall traffic of all the applications. These statistics are also separated in download and upload. We noticed that SOPCast is by far the application producing more signaling traffic, whereas PPLive generates much less signaling than the others. In all the four cases the amount of signaling traffic we sent is much smaller than that we received. This can be explained by observing that we sent a large quantity of video chunks.

Looking at the packet rate for each of them and for both upload and download directions, figure 3.6 shows that the video upload traffic achieves the highest rates. This is consistent with the fact that our host provides the video to several other peers because it is equipped with a fast and symmetrical Internet connection. Moreover, we can sate that SOPCast generates a packet rate higher than all the other applications,

Table 3.4: Signaling Traffic Ratio

	PPLive	**PPStream**	**SOPCast**	**TVAnts**
Total	4.1%	13.6%	19.3%	10.2%
Upload	2.2%	10.8%	13.6%	7.8%
Download	19.2%	25.8%	48.5%	18.0%



(a) PPLive (b) SOPCast

(c) PPStream (d) TVAnts

Figure 3.6: Packet rate of video and signaling traffic in download and upload directions (bin duration is 60s).

especially for video traffic. This application, however, suffered from a large period of time in which the video was not visible. During the same time period the other running application (i.e. PPStream) was properly working. That is why we attribute this behavior to the main source of content and not to the network. This phenomenon happened, at different timings, with almost all SOPCast traces we analyzed. Therefore, we chose to keep this trace for our analysis to allow comparison with the other ones from the three other applications.

3.4 Scaling Behavior of the Traffic

In this section we analyze the collected traffic at different time scales. To this end, we compute the energy spectrum of the traffic at different time scales using a wavelet based

transform method [85]. The smallest time scale we consider is related to 20 milliseconds intervals, as we observed from IPT distributions that IPTs below this value are not so frequent to populate packet-count bins of smaller intervals producing useful packet-rate information. In each interval, we count the number of packet arrivals in both directions (i.e. upload and download). We only count arrivals of packets with data payload and do not take into account empty TCP packets (e.g. Acknowledgments, etc.).

The analysis is carried out by using Logscale Diagram Estimate (LDestimate) [86], which is based on the discrete wavelet transform and allows analyzing the scaling behavior of the packet traffic. LDestimate produces a logarithmic plot of the data energy spectrum, the X-axis of which represents time scales (in octave) of the packet arrivals. Since our bin width is 20 ms, the octave j means the time scale $t = 2^j * 20ms$. LDestimate allows us to visually observe some traffic properties. In the produced diagram, a bump in the energy spectrum indicates a possible periodic behavior of the traffic, a constant energy spectrum a possible memoryless process, and a linear increase indicates a possible long-range dependence.

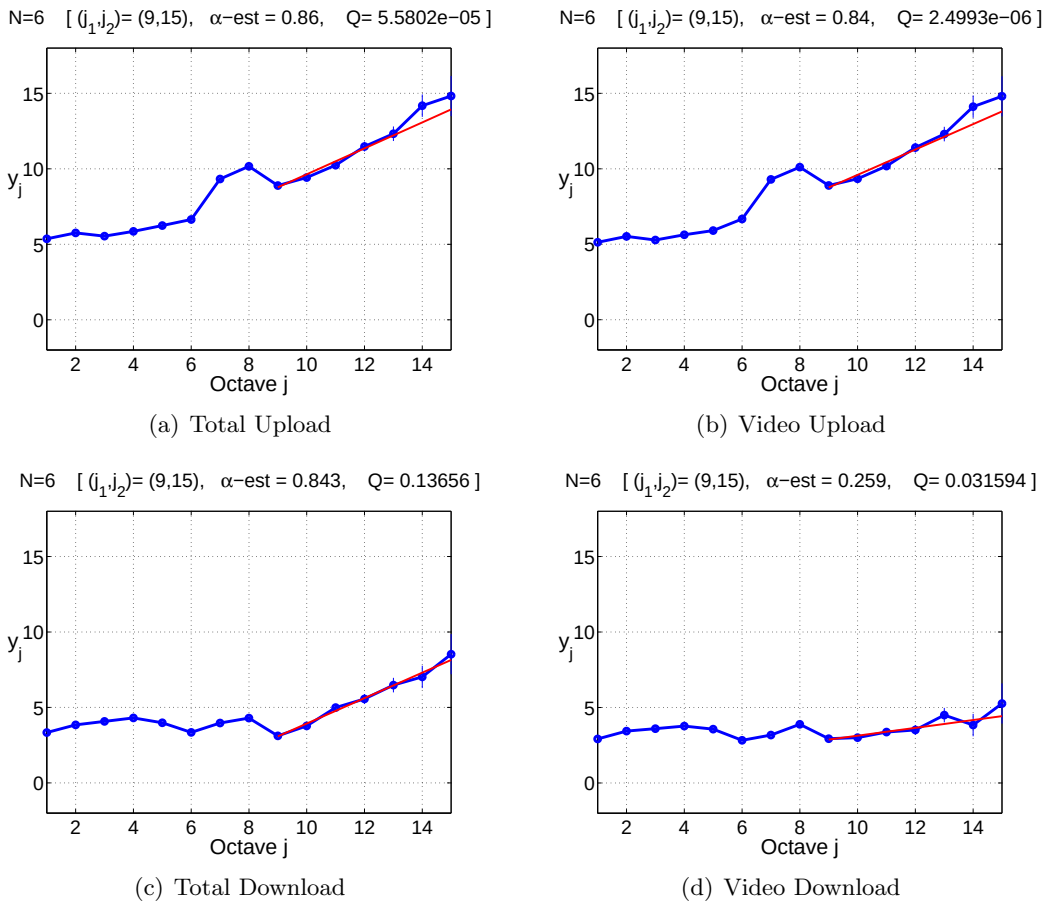
For each application, we separate the traffic in upload and download, and in video and overall traffic (by using the filtering heuristic presented in section 3.3). Therefore, for each application, we obtained four distinct plots: overall upload traffic, video upload traffic, overall download traffic and video download traffic. Figures 3.7-3.10 present the energy spectra on a logscale graph for PPlive, SOPCast, PPStream and TVAnts, respectively.

As shown in Table 3.1, three of the measured applications make extensive use of TCP (PPLive, PPStream and TVAnts) whereas only SOPCast uses mainly UDP. We will refer to an application mainly using TCP as *TCP application*, and *UDP application* for an application using UDP. In the following, we will first present traffic differences between TCP and UDP applications and then highlight the impact of the signaling traffic.

3.4.1 Differences between TCP and UDP Traffic

For the TCP applications, the two upload energy spectra look similar for all the time scales, while the two download energy spectra look similar only until $j = 9$. Moreover, the upload energy spectra of TCP applications are different from their download energy spectra. Furthermore, the TCP applications have similar energy spectra for the correspondent kinds of traffic and direction (e.g. overall upload energy spectra, video upload energy spectra, etc.).

For the UDP application (i.e. SOPCast), figure 3.9 shows that the four energy spectra look similar for all the traffic directions and kinds (the slight difference for the video download energy spectra, will be explained in the next section). Furthermore, they are different from the correspondent traffic of TCP applications. In particular, we can observe that only the TCP applications present an energy bump when the time scale is equal to about $j = 8$ (i.e. $2^8 * 20ms = 5.12s$). Such a bump is more clearly pronounced in upload traffic than in the download one, and it may indicate a possible periodic behavior at these time scales.



(a) Total Upload (b) Video Upload

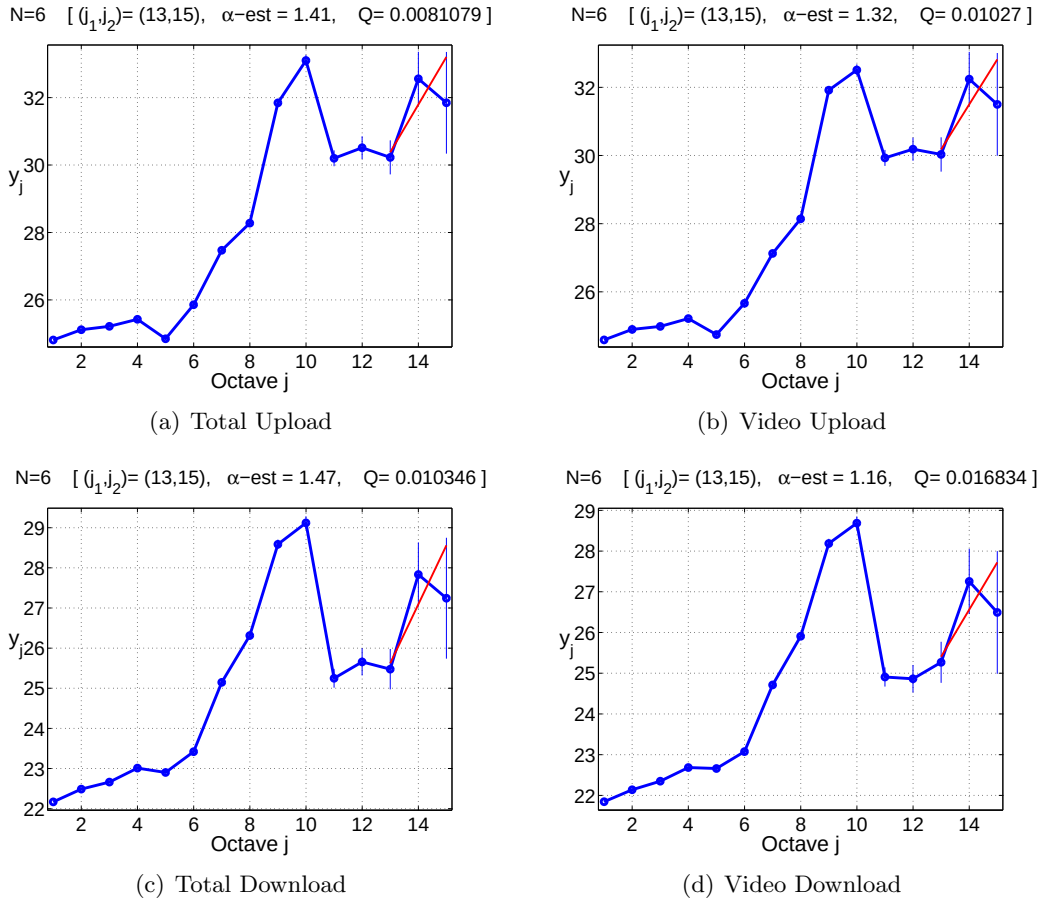
(c) Total Download (d) Video Download

Figure 3.7: PPLive: Traffic energy spectra (bin width is 20ms).

The well-known TCP mechanisms could lead to periodic traffic behavior but not at that time scale, which is a very long period for them. The periodic behaviors could also come from the video broadcast through the network. However, SOPCast does not show any energy bump while it also performs video broadcasting. At present, we are still investigating such behavior because we believe it is an interesting phenomenon and it can indicate how the application design may impact the properties of the generated traffic.

Looking more closely at the energy spectrum for SOPCast, we observe a linear increase whatever the traffic direction or its nature. Therefore, we can state that SOPCast traffic presents long-range dependence (LRD).

LRD means that the traffic fluctuates largely and it is not predictable. In the presence of LRD, it becomes a hard task to provide QoS parameters (e.g. guarantee low and fixed delay, jitter, and packet loss) to users because network conditions are always changing [80].



(a) Total Upload

(b) Video Upload

(c) Total Download

(d) Video Download

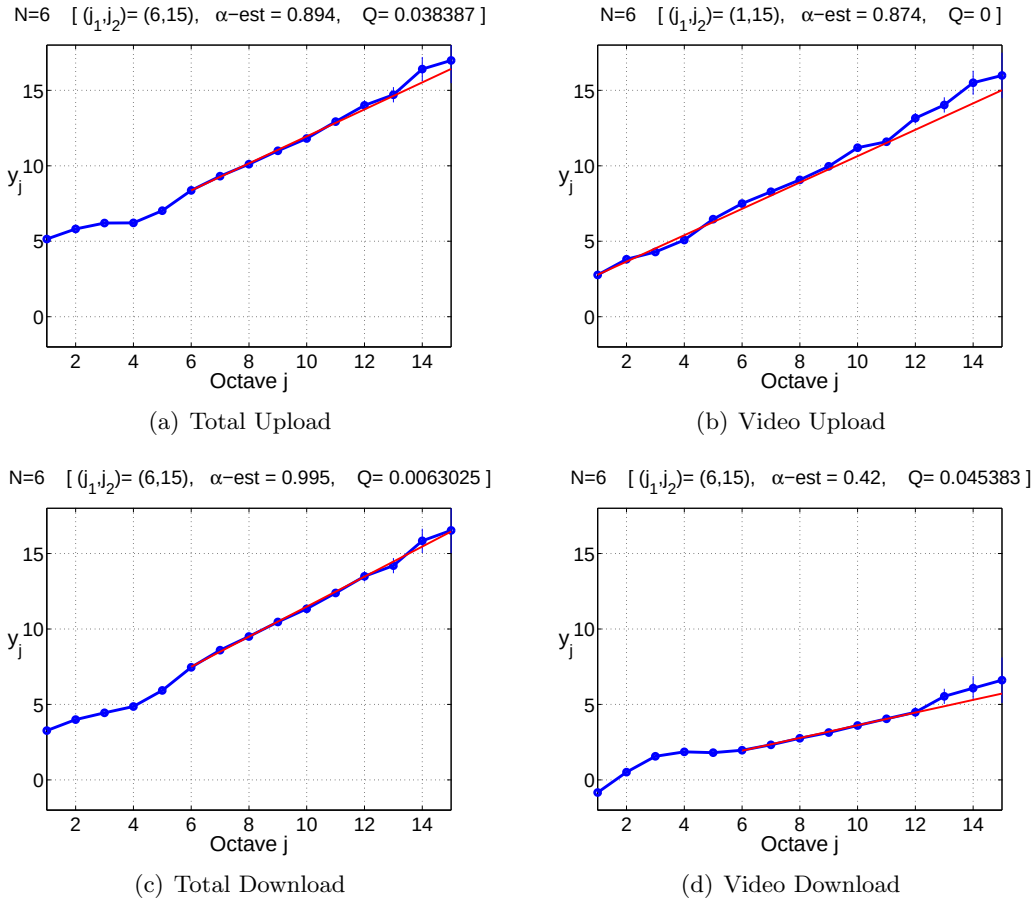
Figure 3.8: PPSTream: Traffic energy spectra (bin width is 20ms).

This also illustrates that the P2P-TV application design impacts (the scaling) properties of the generated traffic.

We summarize the results so far, that TCP traffic exhibits periodic behavior, while UDP traffic has long range dependence. Such results were not evidenced by the time-domain analysis we presented in the previous sections. Moreover, they highlight that the choice of transport protocols for P2P-TV systems is not so trivial. It is usually admitted that the non-elastic data transfer, such as live video, has to rely on UDP but we showed that UDP may lead to traffic LRD. This phenomenon will affect the network conditions and, as a consequence, it will affect the quality of the video stream.

3.4.2 Impact of the Signaling Traffic

For all the applications, whatever the transport protocol they use, their video upload energy spectra look like their overall upload energy spectra. This means that removing



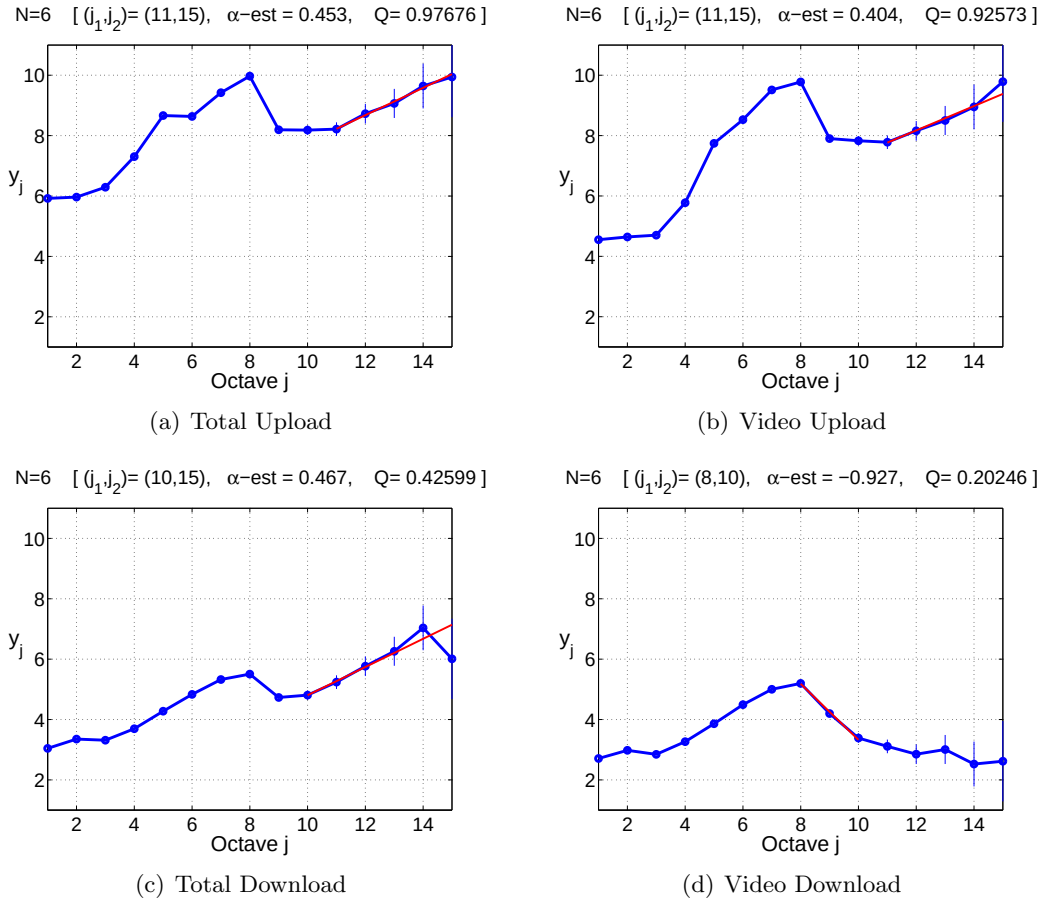
(a) Total Upload (b) Video Upload

(c) Total Download (d) Video Download

Figure 3.9: SOPCast: Traffic energy spectra (bin width is 20ms).

the signaling traffic has no impact on the upload traffic. Instead, as for the download traffic, the video energy spectra are different from the corresponding overall energy spectra, and removing the signaling traffic modifies the download energy spectra. This means that the signaling traffic has an impact on the download traffic but not on the upload traffic.

This observation is important since signaling traffic is necessary to coordinate the data exchange in such P2P systems. And, for scalability reasons, the amount of signaling traffic has to be kept as low as possible. However, Table 3.4 shows that, for all the applications, the signaling is responsible for a fraction of the traffic that is larger for the download than for the upload traffic. Since our node has high bandwidth capabilities, it serves video to many other peers. This explains why the signaling traffic sent by our node to other peers in the Internet counts only for a small part of the overall upload traffic. The download signaling traffic is provided by the other peers to our controlled node which on the other side, just needs to download the video at the video bitrate,



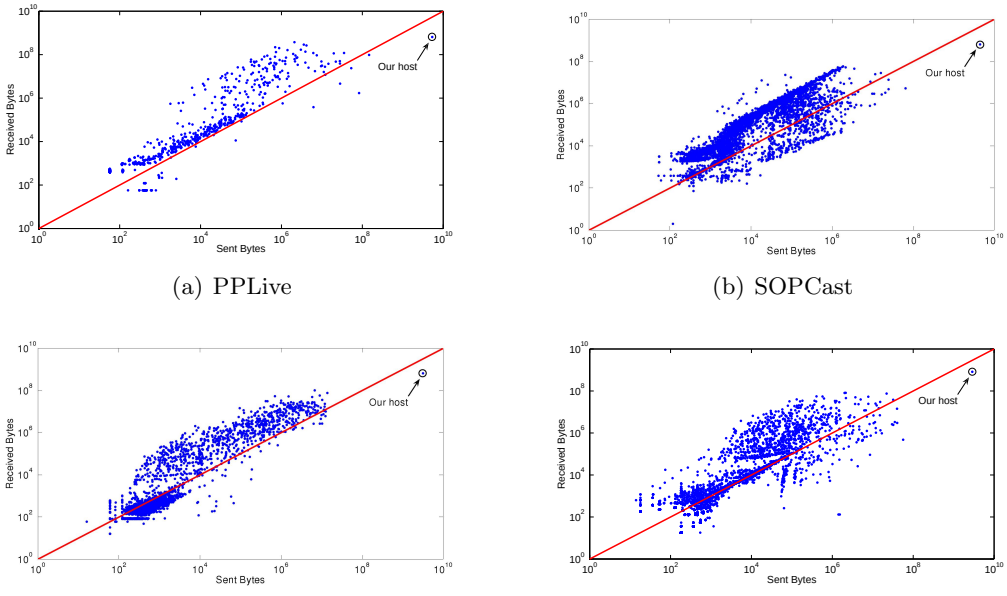
(a) Total Upload (b) Video Upload

(c) Total Download (d) Video Download

Figure 3.10: TVAnts: Traffic energy spectra (bin width is 20ms).

perhaps with some duplicate frames from different sources. The download signaling traffic coming from many other peers therefore counts for a large portion of the overall download traffic. This explains the impact of signaling traffic on the download traffic.

The significant impact of the signaling traffic on the download side implies also that the upload and download traffic have not the same scaling properties and the same impact on the network. The download energy spectra of the studied applications are different from their upload energy spectra. These observations are more relevant since our measurements are made with symmetric access to the Internet. These findings on the different properties of both sides of the traffic and the fact that signaling traffic has a significant impact on the download traffic have to be taken into account carefully when designing synthetic traffic generation models.



(a) PPLive (b) SOPCast

(c) PPStream (d) TVAnts

Figure 3.11: Bytes sent vs. Bytes received for each host.

3.5 Understanding P2P-TV Peer Behavior

In this section we investigate the behavior of the different peers. Our aim is to try to understand the acts of peers and at the same time to spot the similarities and differences among them in P2P-TV systems.

3.5.1 Traffic Generated and Received by Peers

In this section we look at traffic from a peer point of view. Instead of separating traffic in sessions identified by IP addresses, transport protocol, and ports, we consider all the traffic exchanged between each single peer (identified by its IP address) and our host. Table 3.5 shows that during each soccer match, the number of peers that interacted with our host is in the order of a few thousands for all the applications, except for PPLive for which exchanging traffic with less than one thousand hosts was enough to watch the match.

Figure 3.11 shows the number of Bytes sent and received by each peer. In particular, each point represents a peer, and the x- and y-axis represent the amount of sent and received Bytes respectively. The plot has logarithmic scales because the considered values range across multiple orders of magnitude. This plot allows understanding whether the peers receive more data than they send and vice versa. In particular, a point over the bisector (bold solid line in figure 3.11) represents a peer that received more data than it sent, while a point under the bisector is representative of a peer that

Table 3.5: Number of peers interacting with our host. Total number and percentage in the *"Non cooperative region" ((0,0)-(10^4,10^4))* (less then 10KB exchanged in each direction).

	PPLive	**PPStream**	**SOPCast**	**TVAnts**
Total	649	5,956	3,876	5,394
Non cooperative region	305	3,186	3,012	4,075
Percentage	47%	53,5%	77,7%	75,5%

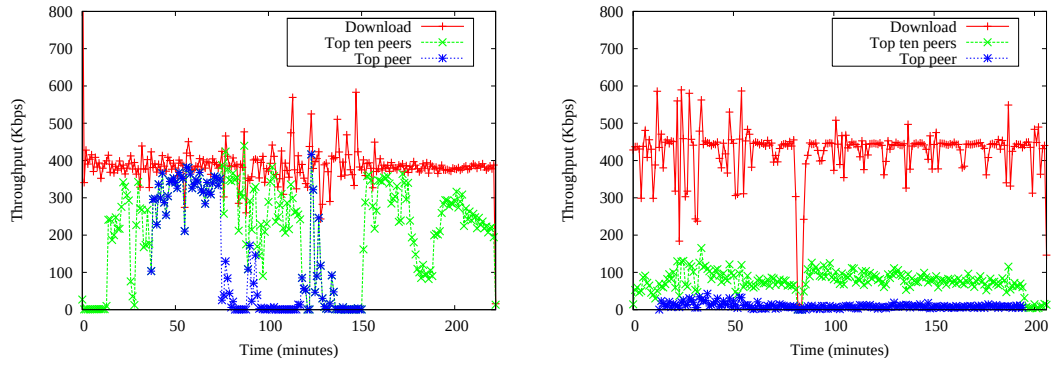
sent more than it received. Clearly, the points on the bisector are related to a perfectly balanced situation.

First of all, we can observe that for all the applications most of the peers are located over the bisector. That is, most of the peers our hosts interact with receive more data than what they send us. This behavior is particularly pronounced for PPLive peers, and for PPStream peers exchanging large quantities of data. This general behavior is due to the fact that our host is provided with a very stable Internet connection and much broader band than what is necessary to transmit a single video. It is therefore used by several other peers to retrieve video chunks. This is also witnessed by the fact that the point in figure 3.11 related to our host is located under the bisector.

Moreover, it is interesting to note that a large quantity of peers is in the lower-left region of the diagrams (<10 KB in both directions), meaning that their interaction with our host is very low. In Table 3.5, to help to interpret the diagrams in figure 3.11, we report the percentage of peers inside the (0,0)-(10^4,10^4) rectangle that we call the *"Non cooperative region"*. Such percentages are quite high, showing that from about 50% to 75% of the peers belong to this area. This may reflect that some relations with peers are poorly utilized and that the created content distribution networks suffer some kind of instability and overhead. It is also worth noticing that the worst results are related to the two applications using UDP.

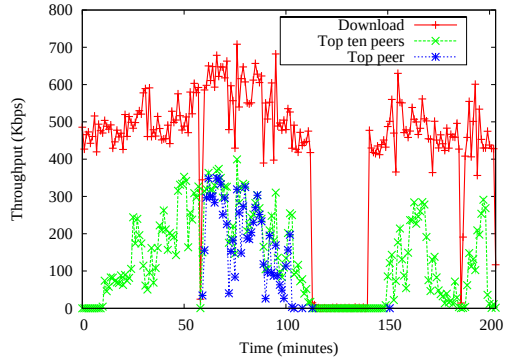
In order to understand the download policies, we computed the amount of data that our nodes downloaded from each of the other peers. We isolated the traffic of the top-ten peers (peers that sent the largest amount of data to our nodes across the entire trace duration), and also the top-peer traffic (top peer belongs to the top-ten peers). In figure 3.12 we plot the total traffic we downloaded, the aggregate traffic downloaded from the top ten peers and that we downloaded from the top peer. Each point of the figure represents a 60 seconds interval (i.e. bin duration is 60s). As said, SOPCast (figure 3.12(c)) received no traffic from minute 130 to minute 140, we watched a black screen during this period. The problem did not occur for network problems because PPStream was working well during the same time period. Therefore, the video source has probably suffered technical problems.

The figure 3.12 shows that the download policies for all the applications are dif-

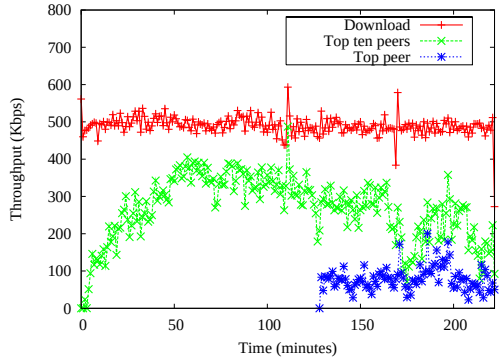


(a) PPLive

(b) PPStream



(c) SOPCast



(d) TVAnts

Figure 3.12: Video download policies: total traffic, top ten peers traffic and top peer traffic (bin duration is 60s).

ferent. For PPLive (figure 3.12(a)), the top-ten peers contribute to a major part of the download traffic and the top peer contributes to almost all the traffic during its session duration. However, such duration is quite short with respect to the entire trace duration. These observations suggest that PPLive gets the video from only few peers at the same time, and switches periodically from one peer to the other.

PPStream download policy is the opposite. For PPStream (figure 3.12(b)), the top-ten peers do not contribute to a large part of the download traffic and neither does the top peer. PPStream has to get the data from many peers at the same time, and its peers have long session duration.

SOPCast top-ten peers (figure 3.12(c)) contribute to about half the total download traffic while the top peer contributes to all of the top-ten peer traffic during its session duration. In a way, SOPCast download policy looks like PPLive policy: it switches periodically the provider peer. However, while SOPCast seems to always need more than a peer to get the video, for PPLive a single peer can be the only video provider.

The TVAnts download policy (figure 3.12(d)) seems to mix PPStream and SOPCast

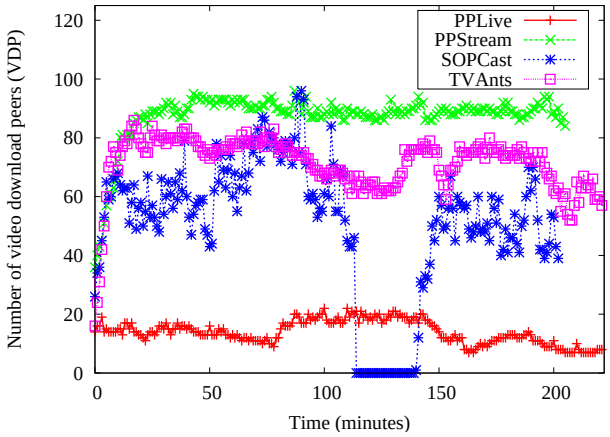


Figure 3.13: Peers neighborhood for all the applications (bin duration is 60 seconds).

ones. For such applications, the top-ten peers contribute to about the half of the total download traffic (like SOPCast), but the top peer does not contribute to a large fraction of that traffic (like PPStream). TVAnts top peer contributes to the overall traffic more than PPStream one even if the former features a shorter session duration.

If we summarize our observations, the presented applications implement different download policies and do not expect peers to have the same capabilities. Some download policies expect peers to stay in the network for a long time (like PPStream) or a short time (PPLive, SOPCast), or expect a peer to have very broadband Internet connection to send all the video (PPLive) or a low one (PPStream and TVAnts). According to the application, a peer can get the video from only few or from many peers at the same time, and its session duration can be various. Different download policies imply different policies for establishing and maintaining connections with other peers (i.e. for handling the peer neighborhood) in order to get the video. This will be pointed out in the next section.

3.5.2 Peers Neighborhood and Session Duration

In swarming P2P systems, peers have to maintain knowledge of their neighbors in order to get the data chunks from several peers at the same time. In figure 3.13 we plot – for each application– the neighboring video download peers maintained by our nodes during the entire trace duration. A neighboring video download peer is a peer which has sent video to our controlled nodes. In the following, we will refer to the number of such peers as VDP (Video Download Peer).

PPLive maintains a relatively low and constant VDP whereas PPStream has a high and constant VDP. SOPCast VDP can be as high as PPStream one but it fluctuates largely. As expected, SOPCast has no VDP when our node receives no traffic. TVAnts VDP is high and also fluctuates.

All the applications maintain a different number of neighboring peers, which corrob-

orates the fact that the applications have different download policies to get the video. As expected, there is a large set of steady peers for PPStream and only a reduced set for PPLive. SOPCast and TVAnts have high and fluctuating VDP. This can be due to the fact that such applications use UDP for part of the traffic (Table 3.1). The VDP fluctuations may come from the non-reliability of UDP, which causes more packet losses and forces peers to keep its VDP always evolving to get the video. This hypothesis regarding UDP may also be strengthened by what we have found in the previous subsection: applications using UDP have by far a larger number of interacting peers with which our nodes do not exchange more than 10KB per direction in total.

In P2P-TV, end-hosts are responsible for relaying flows to each other. End-hosts are not entities dedicated to stay in the network all time: they can join or leave the network whenever they want and are prone to failures. P2P-TV systems have to deal with this phenomenon, also called *churn* of peer. Basically, a high churn of peers will occasion additional delays or jitters for packet delivery, which will affect the video quality. For video live streaming, the churn of peer is challenging because live video has to respect play-out delay to achieve smooth rendering. Here we focus on the video-peer session length (*lifetime*) to study the churn of peers.

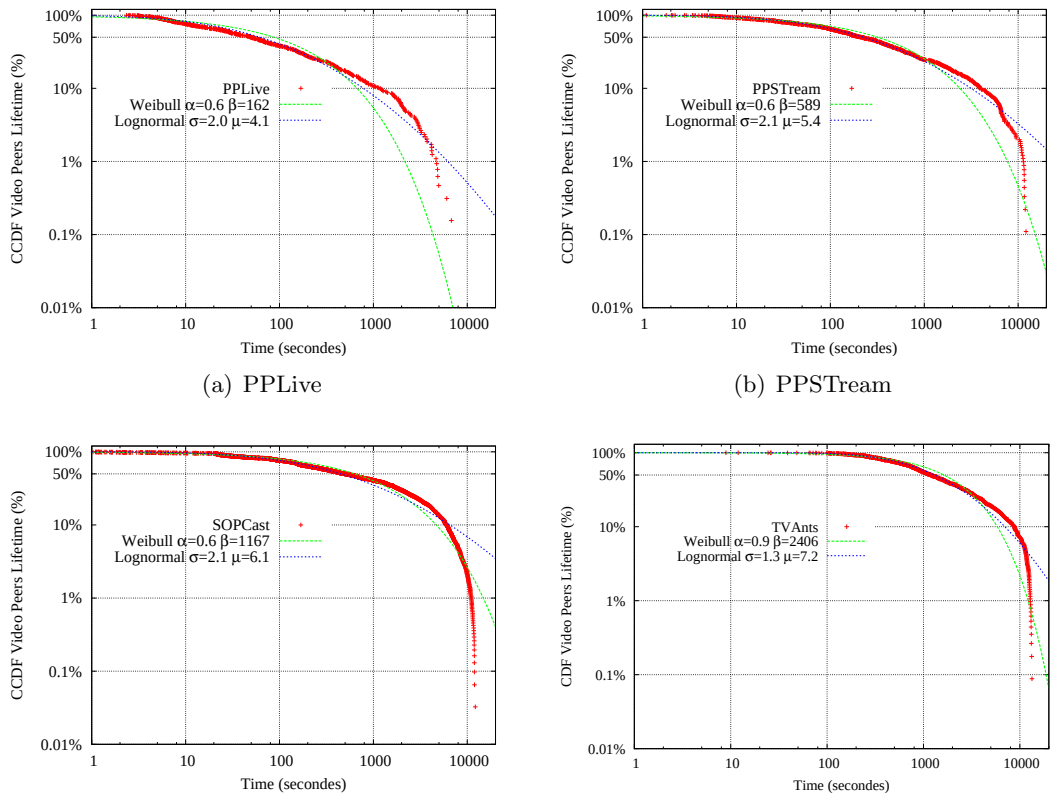
Since our nodes have only a local view of all the peers in the network, we define the video peer's lifetime as the duration between the first and the last time our controlled nodes exchange video traffic with another peer.

For all the applications, the figure 3.14 plots the complementary cumulative distribution function (CCDF) of the video peer lifetime. The average peers' lifetime of all the applications is also presented in the table 3.6.

The average video peer lifetime is different for all the applications and range from 393 seconds for PPLive to 2,778 seconds for TVAnts. From all the applications, PPLive has the shorter average peers lifetime and we observe that about 50% video sessions are shorter than 50s seconds and 90% are shorter than 20 minutes. To put into perspectives, TVAnts has the largest average peers lifetime and 50% sessions are shorter than 20 minutes and 90% than 150mn. 50% of the video sessions are shorter than 4mn for PPSTream and 8mn for SOPCast; 90% of the sessions are shorter than 68mn for PPSTream and 100mn for SOPCast. For all the applications, there are no more than 10% of peers that stay in the network during the entire experiment.

The video peers lifetime and its average vary largely for all the applications but most of the video sessions are short and this is especially the case if we compare with the entire duration of a match (about 120mn).

This fact is very different from P2P file sharing systems where users could stay hours or days in the network to complete the file download [87]. Several reasons can explain these differences between file-sharing and video streaming. Since P2P file sharing distribute files that will be consumed after the download complete, users are more likely to go away from their computer while it continues to download content, and the session lengths will be long. For video live streaming, video streams are consumed on the fly and if the user is away, he has no interest to keep its application running compared to file-sharing; an user benefits from the application only if he is actively



(a) PPLive (b) PPSTream

(c) SOPCast (d) TVAnts

Figure 3.14: Video peers lifetime for the measured applications

present watching the video on its computer. Hence the session times are shorter for video streaming than for file-sharing.

The behavior of users could also explain the short video session length. Even though we focus on a major sport event attractive for users, one can imagine that an user just need to watch briefly the score of the game and leave immediately the network and do not spend two hours watching the game. Since some games were broadcasted on the afternoon, it can be the possible behavior of users watching the score of the game at the office. In the same way, some users like to switch often from one channel to another as this is commonly the case with classical television (channel zapping).

The application mechanisms could also shorten the video sessions lengths, if they force peers to switch from a video provider peer to another one. For instance, in the previous section we observed with the download policy of PPLive that peers may switch periodically from one peer to another to get the video.

Finally, the applications characteristics, their usage, or the behavior of users will have an impact on the video session lengths. The short video peer lifetimes point out the inherent change in partnership between peers during all the video session and emphasize a high ratio of churn.

Table 3.6: Complementary Cumulative Distribution Function of the video peers lifetime (CCDF) and average peers lifetime.

	Lognormal $0.5 * erfc[\frac{log(x)-\mu}{\sigma*\sqrt{2}}]$	Weibull $e^{-(x/\beta)^{\alpha}}$	Average Peer lifetime (s)
PPLive	$\mu = 4.1, \sigma = 2.0$	$\alpha = 0.6, \beta = 162$	393
PPStream	$\mu = 5.4, \sigma = 2.1$	$\alpha = 0.6, \beta = 589$	1,222
SOPCast	$\mu = 6.1, \sigma = 2.1$	$\alpha = 0.6, \beta = 1,167$	1,861
TVANTS	$\mu = 7.2, \sigma = 1.3$	$\alpha = 0.9, \beta = 2,406$	2,778

Since the video peers' lifetime is an essential parameter to model the arrivals and departures behavior in simulations of P2P-TV systems, we model the video peer's lifetime with Weibull and Log-normal distribution. We chose these distribution because they are commonly used to represent life data and are usually called as lifetime distributions [88].

We used easyfit [89] to compute the parameters of these distributions. The fits of the curves are presented with the distributions on the figure 3.14 and the parameters are presented on the table 3.6.

These two curves match well the video peers' lifetime until the last 10% (the right-most region of the plots). Then, the Log-normal seems to fit better the last 10% for PPLive and PPSTream and the Weibull for SOPCast and TVAnts. The last 10% sessions are for peers that stay for a very long time in the system and some of them could probably be the source server and its mirrors. Moreover, This right-most region of the plots is also more likely to show measurement artifacts due to the duration of the experiment. Indeed, it limits the possibility of observing peers' lifetime that last longer than the experiment. Thus, as it is often the case for distribution fits [90], one can neglect the last inaccurate part of the distribution and focus only on the main interesting part which represents the large portion (90%) of the video peers' lifetime and is well fitted by the curves.

By fitting the video peers' lifetime distributions with two curves, we want to emphasize that the goodness of the fit is an open problem. Here, we just aim to provide realistic parameters to model the video peers' lifetime.

A model can also help to understand the properties of the studied phenomenon (video-peers' lifetime in our case). For instance, we will focus on the Weibull distribution because we can more easily interpret the meaning of its parameters.

Clearly, for the Weibull distribution of all the applications, we observed that the shape parameter α is inferior to 1 which would suggest that the probability that peers leave decrease over time. As an example, for PPLive, 50% of peers leave before 50s but the remaining peers can stay until thousands of seconds in the network. It corroborates our intuition on the behavior of users: some users just switch from channels or watch briefly the score and leave while other stay and watch the game for a longer time. In

addition, we can notice that the shape parameter is similar for PPLive, PPSTream and TVAnts ($\alpha = 0.6$) while it is larger for TVAnts ($\alpha = 0.9$). Perhaps this trend comes from the internal mechanisms of the applications, or some other phenomena and it should be deeper investigated in the future.

Regarding the scale parameter β, It differs largely for all the applications and reach its smaller value for PPLive ($\beta = 162$) and the largest for TVAnts ($\beta = 2,406$). If β is large then the distribution will be more spread out and if β is small then it will be more concentrated. The model captured well what we actually observed with PPLive since its video peers' lifetime distribution is concentrated : 90% of its sessions are shorter than 20 minutes and the average video session is short (393s). Differently, for TVAnts the video peers' lifetime distribution is spread out: 90% of the session is shorter than 165mn and it shows the largest average session lengths (2,778s).

The analysis of the distribution parameters could help to characterize the behavior of users watching such sport events. In the future, interesting points would be to study these parameters while different events are broadcasted such as movies or news. Differently from soccer games, these events do not exhibit the same live interest to be watched or have various durations. They could probably show different values for these parameters or the mean of the distribution.

From the study of the video session lengths, we uncover the fact that the characteristics of the applications and the behavior of users lead to short video peers' lifetime. It causes a higher level of churn in P2P-TV systems compared with P2P file-sharing. We fitted the video peers' lifetime with Weibull and Log-normal distribution to model the arrival or departures of peers. The use of realistic workload parameter will help to improve the accuracy of future models and simulations. This could also be useful to add realistic lifetime-based mechanisms to improve the partnerships between peers [91]. We also put our data set available online for the research community [77]; anybody could therefore investigate the traces and extract the exact values of the video peers' lifetime for more realistic parameters.

3.6 Summary

In this chapter, we presented our P2P-TV measurement experiment and we analyzed the network traffic generated by four of the most popular P2P-TV applications. Such applications use proprietary unpublished protocols, making their study challenging. However this work leads towards improving knowledge of current P2P-TV systems. The results here presented can be useful in several fields: (i) to identify traffic generated by such applications; (ii) to understand the impact of their traffic on the networks; (iii) to build realistic simulations and emulations.

To summarize our findings, we outlined similarities and differences among P2P-TV applications in terms of the transport layer protocols and the related ports they use, deriving some interesting properties, e.g. which applications run only on TCP and which ones rely also on UDP, and showing that such traffic cannot be identified by using port numbers.

The first step to understand and identify P2P-TV traffic is to discriminate between signaling and data traffic. We discovered several properties of the traffic that strongly covalidate, for all the applications considered, a heuristic criterion (previously proposed in literature only for PPLive and with slightly different parameters) to separate signaling and data sessions. This step was fundamental to further analyze operation and exchange of traffic in P2P-TV. Moreover, we gained some knowledge regarding statistical properties of this traffic (e.g. Packet Size - Inter Packet Time distribution (PS-IPT), recurring PS, etc.)

Furthermore, this study allows understanding how traffic from peers participating to a P2P-TV network is divided into upload and download, signaling and data. Looking at packet size statistics, packet-rates, and scaling properties of this traffic allows building better simulations and understanding better the impact of these applications on the network. For example, by studying the scaling properties of signaling and data traffic we discovered that for one of the considered applications (i.e. SOPCast, the only one that mainly runs on UDP) there are evidences of long range dependence.

Looking at the peers interacting with our controlled nodes, we also inferred some knowledge regarding peer behavior in P2P-TV. This has also been possible thanks to the use of traces collected during a major event which attracted a large number of peers. We derived information on the network of peers distributing real-time content, noticing that the number of peers with which our nodes were able to exchange significant amount of data was quite low. This was especially true for applications using UDP (we introduced the concept of the *"Non cooperative region"* quantifying the number of peers not contributing significantly to content distribution but rather responsible for an increase in the overhead). Moreover, we found out that, in general, the amount of data sent to each peer by our nodes was larger than the amount received. We also studied the behavior of the top video downloaders from our nodes and the evolution of peers' neighborhood in time for all the applications. Thanks to that, we gathered useful insights on peers' behavior in terms of traffic contribution, stability and robustness of the content distribution networks. We found that applications using UDP present some properties that may reflect less stability and more overhead in the management of peers. We also inferred different download policies used by the applications, revealing a different design of the considered architectures. Finally, we analyzed the durations of the peer connections and aimed to model the distributions with Log-normal or Weibull functions. This information is useful as a reference model for simulations.

Due to the lack of realistic models for P2P traffic [92], simulations could lead to wrong results. Thanks to the results shown and discussed in this chapter, simulations of new architectures can be run using more realistic input parameters. In addition, to the best of our knowledge, no other studies on P2P-TV traffic take into account the fact that the two traffic directions (upload and download) present different characteristics. We believe that, while the results of our study can be useful for both network operators and application developers to understand the behavior of current P2P-TV applications and users, such applications are rapidly expanding and they should be therefore constantly monitored.

Chapter 4

P2P-TV Experiment on Both Sides of The World

In the previous Chapter, we present our P2P-TV measurement experiment and traffic analysis. These experiences have been conducted in order to get knowledge of these systems. As stated before, on the one hand, there have been a lot of architecture proposals that have never been implemented nor deployed on the Internet. It was therefore difficult to assess the feasibility of these architectures on the Internet. On the other hand, the commercial applications are proprietary and do not provide any information about their architecture or implementation. Thus, we aimed to fill the gap between the various architectures proposals that are not implemented and the commercial applications. In addition, the increasing popularity of P2P-TV and its high resources consumption are expected to stress the Internet traffic. Our measurement experiment also helped to study the impact of the P2P-TV traffic on the network.

Among our presented results, we discussed in details about the traffic of P2P-TV and its characteristics. We uncover the main properties of these systems: provide information about the traffic exchanged by peers, the behavior of peers to download the video or their lifetime in the system.

In parallel with other measurement studies [84, 93], our works have been among the first to measure P2P-TV systems to deduce their main mechanisms and characteristics; Since then, P2P-TV measurement studies have become an important research field and there have been many other measurement experiments and traffic analysis on P2P-TV.

Following our first fruitful measurement campaign presented in the chapter 3, we wanted to extend the scope of our study. Indeed, our measurement experiment has been conducted with a single measurement point, collecting one trace of traffic for each application and TV program (i.e: a soccer game). However, the measured applications are used worldwide. The different location of peers, the users' interest for the broadcasted content according to their geographic area or their Internet access environment could have an impact on the behavior of users and the measured properties of P2P-TV.

For instance, a soccer game opposing two European teams could be attractive for many users situated in Europe but suffers a lack of popularity for users situated in other continents such as Asia. Nonetheless, the worldwide popularity of some sports such as the soccer, or some sport teams (such as national soccer team of Italy or English Premier League team of Liverpool) could generate interest for users whatever the geographic area they are.

To this end, we performed a second measurement campaign at a larger scale in order to obtain a more global view of the P2P network. Instead of measuring a P2P-TV with a single measurement point, we measured them with multiple measurement points located on both sides of the world: in France and in Japan.

Among interesting topics, this second experiment will allow us to study the organization of peers in the network, the amount of traffic they exchange and the overall collaboration of peers. We will also pay attention on the distance between peers, the geographic location of the users and the effect of the content on the P2P-TV population. This will allows us to state on the two main mechanisms for P2P systems: the incentive mechanism to enforce peers to collaborate in the system, and the use of mechanisms to keep the traffic local. As we will explain later, the locality of the traffic is an important concern for the Internet Service Providers (ISPs). The collaboration between peers is a an essential conditions for P2P systems to deliver content at large scale. Moreover, by measuring the traffic in two different countries, we were expecting to observe differences between the traffic collected in France and the traffic collected in Japan. In the next section we will explain more precisely our motivations with this second measurement experiment.

As a first part of this chapter, we will present the related P2P-TV measurement experiments that are complementary with our works. Then, in the second part of this chapter, we will present our second measurement experiment of P2P-TV performed on both sides of the world. Based on our findings, we will propose a new incentive mechanism that matches our observations of real P2P-TV systems.

4.1 Motivation

The Internet is becoming the main medium to deliver content to users and the P2P model has shown its ability to deliver content at large scale by involving the end-host to contribute with their resources to forward the data. The P2P traffic is therefore an important concern for ISPs since it has become very challenging to predict the load of the P2P traffic in order to provision efficiently the resources of their network.

When ISPs provide directly content to their customers with common “triple-play” offers including phone, television and video-on-demand services, they can potentially forecast the load of the traffic in their network. Indeed, it depends approximately on the number of users and their usage. Moreover, the traffic stays local within the ISP networks: the traffic is transmitted from the content servers of the ISPs to their customers.

In the case of P2P applications, the content are exchanged between users from

different ISPs. It can results in a situation where a peer in France exchange the data with a peer in Australia, while the same data may be available in a closer peer in Germany, or even in the same network in France. It has been shown that the P2P traffic crosses the networks boundaries multiple times [94]. This is an important issue since the bottleneck of the Internet is assumed to be either in the access network or on the links between ISPs, but not in the backbones of the ISPs. This traffic could therefore overload the links between ISPs. The P2P traffic is a dilemma for ISPs since it can be exchanged with many different ISPs, without any knowledge on the amount of traffic and its nature. In the case of video streaming, this traffic is non-elastic, consumes a large amount of bandwidth and is sensitive to network delay. The large presence of P2P traffic consumes the network resources and is potentially costly for the ISPs.

The locality of P2P traffic is an important issue for ISPs. It is also important for users because long distance could lead to long end-to-end delay. For video traffic, long delay to get the video may disturb the quality of the video. In addition, the P2P applications rely –by nature– on the reciprocal contribution between peers.

In order to evaluate P2P-TV applications, it is therefore important to study the locality of peers and their distance in the network, but also the amount of data the peers exchange and their level of collaboration. The locality of peers is relevant for ISPs but also for users. The reciprocal exchange of data between peers is of interest for users; It is also an essential condition for the viability of P2P systems to deliver content at large scale. To this end, our second measurement experiment will focus mainly on these topics.

4.2 Related Measurement Experiment of P2P-TV

Since our previous measurement experiment, there have been an increasing number of P2P-TV measurement studies. Some of them use a similar approach to our and perform passive measurement of the network traffic generated by popular P2P-TV systems. We chose this approach because we did not have access to the measured systems, which appear to us as “black boxes”. By measuring the P2P-TV traffic on the Internet, we also wanted to obtain measurement under realistic conditions.

Differently, some studies perform active measurements of the systems by actively probing the peers of the network. It can help to get more precise information of the system but the probes could also disturb the behavior of the applications and the measurement experiment. Other studies perform their experiments on controlled testbeds such as PlanetLab [95]. PlanetLab is a research community testbed deployed worldwide, allowing to perform experiment on a controlled environment. Moreover, some studies have been conducted by the P2P-TV designers themselves in order to present their system, to evaluate and improve it. They have therefore a complete knowledge on the system under test.

In the following of this section, we will present the main P2P-TV measurement studies that we classify into three different categories: measurements with global knowledge of the system, active measurements and passive measurement of the system.

4.2.1 Measurement with Global Knowledge of the System

In this approach, the P2P-TV architecture and its implementation details are known a priori (source code), and no assumption is needed to understand the behavior of the systems. To obtain such level of details, these studies have been conducted jointly with the application designers themselves [76, 96, 97], or the P2P-TV companies have made their data set available for joint research in order to analyze, evaluate and improve their system which is continuously monitored. Typically, the companies keep track on the peers of the systems with server logs and periodic peers reports.

CoolStreaming [12, 69] is the implementation of DONet, an architecture that have been largely presented in the chapter 2. A prototype has been experimented on PlanetLab [95]. it allows having a complete control on the measured environment and the system. By experimenting on PlanetLab, the designers were able to control the number of peers in the network, or to choose the peers' lifetime or their arrival time in the network. Thus, network statistics such as user's behavior in the whole system and the quality of video reception are provided, and also a detailed study on the control overhead generated by this system.

This measurement study with complete knowledge of both the system and the environment is very interesting but present also some limitations: the results cannot be extended to the Internet environment because the PlanetLab testbed differs too largely from it. For instance, the PlanetLab testbed allows to have only hundred of peers in the network whereas an experiment on the Internet may involves thousands of peers or more. As it is commonly reported with the feedback of PlanetLab users, the PlanetLab nodes are prone to failures because of technical reasons, leading to difficulties to test or reproduce the experiments on this testbed. Moreover, the PlanetLab testbed is mainly composed with nodes situated in academic network. These networks are well managed by Universities and are likely to be well connected to each other differently from regular nodes on the Internet. Moreover, P2P-TV applications are used by end-users with residential access to the Internet. The end-users network conditions (asymmetric access) are very different from those provided to the nodes by the PlanetLab testbed (symmetric and high-speed access).

Even though Coolstreaming could be considered as the first successful P2P-TV system, these results cannot be extended to the Internet case. Such kinds of measurement experiments are therefore interesting for the designer to tune their own system and to validate their design choices.

In 2008, Bo et al. presented the new Coolstreaming architecture [70, 71]. From a mesh-based architecture with pull policy, the architecture has been completely redesigned and uses a hybrid pull/push policy with multiple video sub-streams. Again, the new Coolstreaming architecture has been implemented and, this time, the measurements have been performed on the Internet. To this end, each peer reports periodically its activities to a log server (video quality, achieved bandwidth or relationships with other peers). Some statistics on the type of users directly connected to the system or behind NAT or firewall are provided. They show that the use of NAT drastically af-

fects the overall performance of their system since 45% of the users have limited upload capacities because the current system does not use any NAT traversal technique.

This new Coolstreaming implementation is quite new and is not widely used as popular applications such as PPLive or PPSTream. The presented results are thus to be considered with regards to the limited deployment of this application on the Internet.

As another example of measurement study with a complete knowledge of the system, Huang et al. unveiled part of the mystery surrounding the architecture of PPLive [76]. However, they present their Video-on-Demand service (VoD) for PPLive and do not present their live television service. Even though VoD service is out of the scope of our study, there are some common points between these two services such as the main mesh-based architecture, the chunk replication strategy or the use of buffer map to announce the chunk availability. Thus, we chose to mention this interesting study in this related work. Overall, all these services are about streaming content over P2P networks, even if in VoD, there is less synchrony between users that can watch very different parts of the movie at the same time (beginning, middle or end of the movie).

The PPLive system is continuously supervised with each peer reporting precisely its activities to a log server. In [76], Huang et. al evaluate the health of the systems and the satisfaction of users watching three movies with different popularity. Among their results, they quantify the users' behavior, the effectiveness of the replication scheduling strategies, the video session length, the chunks of data available at each user, or the number of users behind NAT.

Zattoo [98] is an American and Swiss company that released their own P2P-TV system in the beginning of 2008. Differently from other companies, which broadcast TV without any regards to the copyright policies, Zattoo established partnerships with traditional TV broadcasters and broadcasts their channel within each country with mutual agreements. This application is currently available in Europe and its success is increasing since it already counts millions of registered users. Zattoo has also been built with monitoring capabilities to measure both system and its performance for design improvements as well as the user satisfaction. Based on the data collected by Zattoo [97], Shami et. al presents statistics about the capacities of peers to redistribute the data and the ratio of peers behind NAT systems. Their study shows that most of the users with Zattoo are behind NAT systems and cannot contribute to the systems. Remind that P2P architectures have been mainly chosen for their self-scaling properties to allow a television service on the Internet (chapter 2), the massive presence of users behind NAT is an important issue for commercial applications. Indeed, if the peers are not able to redistribute data to other, then the system will not be able to scale to a large number of users.

UUSee [18] is a commercial application that uses a set of dedicated servers to provide the video to users and support the P2P streaming topologies. By obtaining the source code of a client from UUSee, Chual et al. [96] modified it by implementing measurement and reporting capabilities to inspect the behavior of the UUSee P2P streaming service. The performances of the source servers have also been continuously monitored. Each peer reports every 5 minutes to a log server statistics, among which their IP address, the channel it is watching, its buffer map, or its list of partners. For this experiment, the

entire data collection covers about a year. This study shows that the streaming quality experienced by peers has been decreasing over time, as the source servers' capacities are saturated. The streaming servers have became the bottleneck of the systems while the peers contribute with their upload bandwidth and constitute the major part of the video transmission. Since some channels are popular with many users and some are not, the server bandwidth allocation should be forecasted within each channel to avoid the server bandwidth bottleneck problem. To this end, based on their measurement of the server performances, they propose a model to predict the demand in each channel and to provision the server capacities according to the demand over time. Even though the proposed model is efficient to predict the current demand in each channel, they replay the same traces to evaluate the model. It is therefore difficult to state whether the proposed model could be used with other applications or under different network condition or users' demand.

Based on traces they obtained from UUSee [91], Liu et. al explore the influential factors to peer longevity and bandwidth contribution level in a large scale P2P live streaming system. By discovering metrics to predict the superiority of peers such as the peer initial buffering level and ISP membership, they propose a *superiority index* to favor these superior peers in the system because these peers would be more likely to contribute in the P2P network. They shows the effectiveness of this *superiority index* in improving the overall stability and scalability of the P2P streaming system. However, they evaluate the proposed *superiority index* by replaying the same traces used to discover the metrics to detect the superior peers. In order to have a fair evaluation of the proposed solution, the *superiority index* should be evaluated by using another set of data different from the initial one.

4.2.2 Active Measurements of P2P-TV

Another approach to infer the main properties of the P2P system is to perform active measurements. To this end, a modified client is used to send probes to the other peers of the network. With these probes, the modified client crawls the entire networks and collects information. For instance, this information could be a list of peers, or their available data. The crawler can then probes again these new discovered peers, even if the normal behavior of the application would have keep these peers unknown from the crawling peer. The use of a crawler can then provide very useful information but it also generates traffic that is not part of the application and could impact the normal behavior of the application.

To our knowledge, the only active measurement studies of P2P-TV systems focused on PPLive. In their works, Hei. et al. [99] or Vu et al. [100, 101] instrumentalized a modified client to probe actively the PPLive network. In order to perform their measurement, it was necessarily to know the exact way of working of a regular client to perform efficient probing. To this end, as a first step, they had to do reverse engineering of a client by measuring and analyzing its traffic and behavior [84]. Then, by actively probing the network, these active measurement studies are more likely to characterize the overlay networks of these systems, or the popularity of different channels. In their

work, Vu et al. [101] indicate that the PPLive overlay characteristics differ from those of P2P file-sharing. For instance, they show that PPLive overlays are similar to random graphs in structure and thus more robust and resilient to the massive failure of nodes.

By getting information on the blocks available by each peer, Hei. et al. was able to study the overall quality of the video for all the users in the PPLive network [52]. They also instrumentalize their crawler to transmit fake video chunks, demonstrating that PPLive was not suited for pollution attacks and propose some solutions to solve this issue [102].

4.2.3 Passive Measurements of P2P-TV

Passive measurements of a system are used when it is the only feasible way to get knowledge of the considered system. There is no detail about the source code and the measured systems appeared as "black boxes". The traffic measurement and analysis will help to infer the underlying mechanisms of these systems and their global properties. As an example, our measurement study presented in the previous chapter falls into this category.

Prior to our work, Hei et Al. [84] was -to the best of our knowledge- the first work to measure P2P-TV traffic. They focus on PPLive, which was the most popular P2P-TV application and perform passive packets sniffing by collecting traffic from peers situated in their campus network with high-speed Ethernet access and in residential network with cable modem. Their results discussed about the video session lifetime of peers or the download rate of both kind of peers: residential or campus. They also provide a brief peer geographic distribution study. In parallel with this work, Ali et al. [93] made passive measurements of PPLive and SOPCast applications and analyzed the performances and characteristics of such systems. They also provide their own methodology to study the data exchanges of such P2P applications and the behavior of peers behind NAT systems.

These measurement studies are very close and complementary with our works; P2P-TV applications are passively measured to infer the underlying mechanisms and to study their global behavior. However, our work is different from these, since we do not focus on a single application [84] or, a couple [93], but on a set of four applications used worldwide. An important distinction between Hei works and ours comes also from the live interest of the measured event. It is intuitive but corroborated by Veloso et al. [103] that traffic patterns have not the same characteristics whether broadcasted content exhibits a live interest for users or not. At the end, our work present a deeper P2P-TV traffic analysis and also provide a scaling analysis to characterize the correlation structure of the generated traffic at different times scales to understand its properties and its impact on the network and a careful analysis of peers behavior in P2P-TV networks.

Among passive measurement studies, some other chose to perform their experiments on PlanetLab to have control on the network environment. The PlanetLab testbed counts hundred of nodes but only a subset of them can be available at the time of the experiment. Even though the experiment on PlanetLab allows having a complete view

of the P2P network, these measurement experiments are limited to small-scale network and can hardly be generalized to the global behavior of P2P-TV on the Internet.

For instance, Tang et. Al [104] experiment the topology dynamics of SOPCast on PlanetLab with 51 to 132 PlanetLab peers. They focus on the SOPCast graph and provide statistics such as the incoming or outgoing degree distribution and the activation period of a neighbor link between peers. In [105, 106], Yu et al. also study the quality experienced by the users by tuning the delay or loss parameters. However, these results come from a small-scale experiment of a P2P-TV system and cannot represent the real behavior of such system at the Internet-scale. Indeed, commercial applications claim to count thousands of concurrent users in the network, orders of magnitude larger than the number of PlanetLab nodes used in this experiment.

Alessandria et. al [107] define a simple experimental setup to analyze the behavior of four commercial P2P-TV applications: PPLive, SOPCast, TVAnts and TVUPlayer. Similarly to our work, they capture the traffic of running application in their campus network but vary the network conditions such as delay, loss and available capacity with their access router to study the ability of the different P2P-TV applications to adapt dynamically to these changes. In this study, they show that all the tested applications are effective in trying to overcome network impairment. However, in case of bandwidth bottleneck for all the peers, their behavior become aggressive to download the video stream, and potentially harmful for the network. This measurement experiment help to understand how these applications react to these parameter changes. However, the testbed and the range of the tested parameters are not necessarily realistic or do not reflect what could really happen on the Internet.

4.2.4 Discussion

In this section, we present the main related works on the measurement of P2P-TV systems. Some studies have a deep level of details because they have access to the source code of the applications and know the underlying architecture to deliver the video streams (i.e. number of source server etc.) Other studies probe actively the network in order to get as much level of details as the approach with a complete knowledge of the system. The main drawback of the active measurements is that the measurement by itself could disturb the normal behavior of the applications and bias the results. In addition, it is a very challenging task to succeed the complete reverse engineering of an application; it needs lots of resources –in term of people or time– to infer and analyze the P2P-TV application. The most classical approach is to perform passive measurement of the system under test in order to infer the main mechanisms, properties and their impact on the network. Our previous measurement falls into this category, in parallel with the works from Hei [84] and Ali [93]. Since the measured systems appeared to be “black boxes”, it is somehow difficult to learn from the system and we have to rely on assumptions. However, the passive measurements of P2P-TV systems do not disturb the normal behavior of the applications and the measurements are, at least in our case, performed under realistic Internet conditions. Thus, our assumptions are more likely to be credible with regards to the really deployed systems.

Finally, the measurement of P2P-TV systems has turned into an important research topic by itself because it has become urgent to have a better understanding of these systems and their potential impact on the networks [108]. Indeed, these applications are expected to be massively used in a near future and will contribute to a significant part of the Internet traffic. However, most of these measurement studies have only a local point of view of these systems since the P2P-TV traffic is only collected on a single measurement point. Moreover, they do not focus on the behavior of users or the worldwide P2P-TV usage. These topics are –among others– the main goals of our second measurement campaign, which we present in the following sections of this chapter.

4.3 Presentation of the Experiment

For our second experiment, we measured the network traffic generated by several popular P2P-TV applications collected at multiple points located on both side of the world: in France and in Japan. Our first motivation was to extend the scope of our previous study and obtain a more global view of the measured P2P systems. By adding more measurement points, our second experiment testbed will allow us to characterize the overlay network, as well as the organization of peers and their exchange of traffic. In this section, we present in details our measurement experiment by introducing our testbed, the measured applications and the experimental process.

4.3.1 Large-Scale Experimental Testbed

For our second measurement experiment, our testbed is distributed on both sides of the world in two distinct locations: in France and in Japan. The French part of the testbed is similar to the previous one with PCs directly connected to the Internet with 100Mbps Ethernet access through our campus network. We added new PCs with 1.8GHz CPU and common graphic card capabilities, and the number of measurement points (PCs, i.e. nodes) vary from three to four nodes depending on their availability at the time of the experiment. As explained before, each PC used Windows XP and run one P2P-TV application and Windump to collect the traffic. This testbed has been reproduced in Japan within the WIDE network and we used two or three PCs directly connected to the Internet with 100Mbps Ethernet access. All the PCs in France and Japan were equipped with standard 100Mbps Ethernet cards.

For clarity reasons, we refer to the PCs situated is France as *France A, France B France C and France D*, and those in Japan as *Japan X, Japan Y and Japan Z*. In the following, we may call our PCs as our *controlled peers* or *nodes*. Basically, we used these terms with the same meaning but we usually use *peers* for the remote peers and *nodes* for our controlled PCs. Our large-scale testbed is illustrated on figure 4.1.

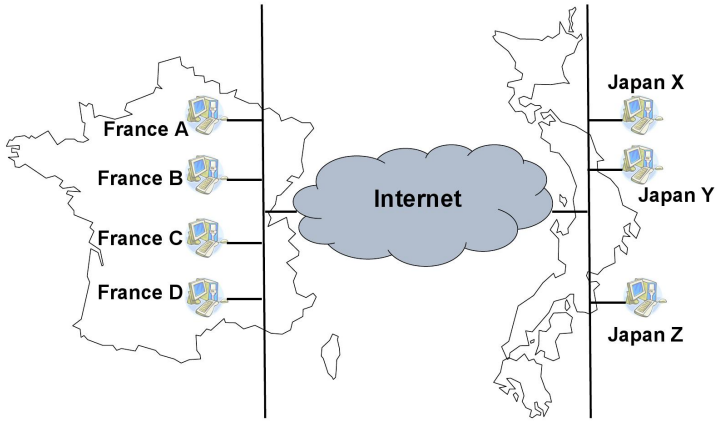


Figure 4.1: Measurement experiments testbed. Each PC is directly connected to the Internet within LIP6 or WIDE network.

4.3.2 P2P-TV Systems Under Test

For the experiment, we still focus on the most popular P2P-TV applications, according to the feedbacks we have from the online community, our personal feedback and the quality we experienced with them. In this chapter, we will present results with PPStream, TVUplayer [19], SOPCast and TVAnts. Compare with our previous experiment, we did not measure PPLive but prefer to include TVUPlayer to our list of measured applications. We decided to remove PPLive from our measurement experiment for the following reasons.

First of all, TVUPlayer has become a successful application, very appreciated by the users; it has also became an application of interest for the research community [107].

Secondly, we are still focusing on live sport event (i.e. soccer games) and, at the time of our experiment, there was no live sport channels available on PPlive to watch the soccer games anymore. When we investigated this point, we discovered that the PPLive users had to find a proxy server in China in order to obtain the live video streams. The use of a proxy server will bias the measurement experiment to collect the traffic. We also believe that a "standard" user will not take time to set up a proxy while other applications are available and efficient to watch the same TV program in good conditions. We discussed this topic largely in some dedicated webforums [75] and the feedbacks regarding PPlive were very mitigated: it appeared to us that the popularity of PPLive suffered from this trend. PPLive is still an application of interest to watch other TV programs but it does not fall into our desired condition for our experiment: to imitate as simple as possible the behavior of a standard user willing to watch a soccer game with a P2P-TV application.

Third, as stated in the previous section, PPLive has became a research topic by itself, with many studies only focusing on this single system [109]. To our opinion, it is not a good idea to focus only on a single P2P-TV system even though it is the most popular one and also one of the first that demonstrates the feasibility of the TV broadcast over P2P networks. It is important to keep our study general with several examples of systems and measurements, and not be closely related to a specific implementation of a system.

Finally, a large part of the PPLive architecture and mechanisms have been revealed because of the important research work made on this peculiar sytem [99] or by their designers [76].

4.3.3 Process of Experiment

To obtain a more global view of the measured P2P-TV network and the overlay characteristics, for each experiment we used our experimental testbed with a single application at the same time.

We still focus on live events and more precisely on sport events because it exhibits an interest to be watched in live, compare to other TV programs such as movies or TV shows. We accomplished our measurement campaign during a three-months period in summer 2007, where we were able to access the WIDE network from the University of Tokyo in Japan. Almost every day, we performed experiments, measuring most of the sports events of this summer with different applications, especially soccer games, but also 2007 IAAF world championship in athletics or 2007 rugby World cup. At the end, we measured about 50 distinct sport events (soccer, rugby or athletic). Since these events have been measured with five to seven measurement points, we collected about 300 traffic traces. At the end, the entire size of our dataset exceeds 1.5 TBytes.

Among all the measured events, we chose to focus on soccer games because there is a large community of users that use such applications for this purpose. From our traces collection, we extracted four measurement experiments representative of our dataset. Each of these four measurement experiments have been performed with a single application at the same time: PPStream, TVUPlayer, SOPCast and TVAnts.

We measured PPStream with five PCs (nodes): three situated in France and the two other in Japan. We collected the traffic on August 15th 2007 during the preliminary round of the UEFA champion's league between Liverpool and Toulouse. The traffic of TVUplayer has been collected with six nodes (three in France and three in Japan). The measured event was a qualifying game for the 2007 Olympic soccer tournament between Japan and Vietnam on August 22 2007. For SOPCast, six nodes were available (four in France and two in Japan) and the event was also a qualifying game for the Olympic tournament between China and Japan. This event took place on August 03 2007. Finally, TVAnts has been measured with all the seven nodes (four in France and three in Japan). This experiment happened during another qualifying game for the Olympic soccer tournament between China and Vietnam on August 23 2007.

It is important to notice that three of the measured events involve Asian countries, where the remaining one involves European countries. China is the largest country

where P2P-TV is already very popular since most of the P2P-TV applications have been released by Chinese companies. Thus, we were expecting that China counts a large number of P2P-TV users. Regarding Japan, it is one of the countries where the measurement experiment took place. It is also the most technologically advanced countries in Asia, and probably in the world. Many Japanese Internet users already have high-speed access with available fiber to the home technology (FTTH) compared with classical residential DSL Internet access [110, 111]. We were wondering if this technological gap between Japan and other countries could have an impact in our measurements. Vietnam is a smaller country with much less users and we believe that there are still poor capabilities to connect to the Internet. However, Vietnam was involved in two measured events. We were wondering, since Vietnam is still an emerging market, if it would show advanced usages of the Internet such as P2P-TV. The last measured event is a common European soccer game in the most prestigious soccer championship. It also involves a famous team (Liverpool) and an open question was if the worldwide fame of the soccer team of Liverpool could be perceived in our measurement studies. Moreover, the other soccer team was from France, which is also a country where the measurement experiment took place. There is no evidence that these facts would have an effect on the collected traffic traces but we would therefore keep in mind this information for our traffic analysis.

The table 4.1 summarizes the collected traces. All the traces have the same duration of 2h45mn (165 minutes). This duration is slightly larger than a soccer game (105 minutes) because we wanted to capture the effect that happens at the beginning or the end of the games.

4.4 General Observations

In this section we will present the statistical properties of the studied traces. Then we will investigate the traces and analyze the traffic in the following sections. Since we are discussing about 24 traces at the same time coming from 4 different applications, we will adopt the following notation to refer to each trace. Each trace will be called *XXX_Yz*, where:
–*XXX* are the three first letters of the application name: *PPS* for PPSTream, *TVU* for TVUPlayer, *SOP* for SOPCast and *TVA* for TVAnts;
–*Y* is the first letter of the country the trace come from: *F* for France and *J* for Japan;
–*z* is the local number of the nodes: a, b, c, d for nodes in France or x, y, z for those in Japan. As examples, the trace collected by France C for PPStream will be called PPS_Fc; the trace from Japan Y for SOPCast will be called SOP_Jy etc.

The table 4.1 shows for all the traces the following information: the entire size of the trace in GigaBytes (GB), the number of packets in the trace, the ratio of upload and download traffic according to the transport protocol TCP and UDP. Table 4.2 shows also the number of IPs present in each trace and other information that we will explain later. As an example, the size of TVU_Fb is 1.6 GB and it counts 4.1 millions of packets. Its upload traffic represents 68% of the overall traffic (i.e. 1.1 GB). TCP

Table 4.1: P2P-TV traces summary. All the traces have the same duration: 2h45 (165mn)

PPSTream	**Size** (GBytes)	**#Pkt** ($\times 10^6$)	**Upload** **Size** (%)		**Download** **Size** (%)	
			TCP	UDP	TCP	UDP
France A	3.1	3.5	82	0.0	18	0.0
France B	3.1	3.5	82	0.0	18	0.0
France C	3.1	3.4	82	0.0	18	0.0
Japan X	3.0	3.3	82	0.0	18	0.0
Japan Y	3.1	3.1	82	0.0	18	0.0
TVUPlayer	**Size** (GBytes)	**#Pkt** ($\times 10^6$)	**Upload** **Size** (%)		**Download** **Size** (%)	
			TCP	UDP	TCP	UDP
France A	1.7	4.4	2.0	64	2.0	32
France B	1.6	4.1	6.0	62	0.0	32
France C	1.6	4.0	3.0	64	2.0	31
Japan X	1.8	5.3	5.0	62	1.0	32
Japan Y	2.2	6.1	3.0	68	1.0	28
Japan Z	3.1	7.8	6.0	72	0.0	22
SOPCast	**Size** (GBytes)	**#Pkt** ($\times 10^6$)	**Upload** **Size** (%)		**Download** **Size** (%)	
			TCP	UDP	TCP	UDP
France A	1.0	2.6	2.0	41	8.0	49
France B	1.2	3.0	1.0	49	7.0	43
France C	1.2	3.0	2.0	51	11	36
France D	1.0	2.6	1.0	40	7.0	52
Japan X	3.2	6.7	5.0	73	0.0	22
Japan Y	3.0	6.4	1.0	76	1.0	22
TVAnts	**Size** (GBytes)	**#Pkt** ($\times 10^6$)	**Upload** **Size** (%)		**Download** **Size** (%)	
			TCP	UDP	TCP	UDP
France A	2.7	3.3	56	23	13	8.0
France B	2.0	2.4	47	25	22	6.0
France C	2.6	3.1	56	22	16	6.0
France D	2.7	3.3	55	24	14	7.0
Japan X	2.4	2.9	56	21	20	3.0
Japan Y	2.3	2.8	51	24	19	6.0
Japan Z	2.5	3.0	57	21	17	5.0

represents only 6% of the trace (i.e. 96 MB), while UDP is massively used in both directions and counts for 94% of the entire traffic (i.e. 1.5 GB: 992MB in upload and 508MB in download).

Based on the tables 4.1 and 4.2, we have the following observations: For PPStream, all the traces are about the same size ($\approx$ 3.1 GB) whatever they are collected in France or Japan. They all present the same bandwidth ratio: 82% for the upload and the remaining 18% for the download. All the PPStream traces count the same amount of packets, even if PPS_Jy counts about 10% packet less than PPS_Fa for the same size. Regarding the number of IPs in each trace, we can observe that PPS_Fa and PPS_Jy counts about 2,500 different IPs while the three other traces count more than 3,000 IPs. This is the only major difference we can observe in the statistical properties of the PPSTream traces. Since the differences are for the two locations (France and Japan), we cannot infer a peculiar behavior for the French or Japanese side.

For TVUPlayer, all the French traces are about the same size ($\approx$ 1.6 GB), whereas the size of the Japanese traces are larger and different from each other. TVU_Jx is 1.8 GB large and it is a similar size as the French traces. TVU_Jy and TVU_Jz are much larger and are respectively 2.2GB and 3.1GB. The number of packets increases with the size of the trace (4.4 to 7.8 millions of packets). All the TVUPlayer traces counts about 1,000 IPs except TVU_Fa which counts twice as much IPs as the other ($\approx$2,000). However, TVU_Fa is only 1.7 GB, which is a common size compared to other French traces but it is much smaller than the Japanese one (2.2 and 3.1 GB). It was interesting to notice that this small-size trace involves much more IPs addresses than the large-size traces.

For SOPCast, the French traces exhibit the same statistical properties: same size (<1.2GB), number of packets (2.6/3 millions) and traffic ratio (about 50–50%). The two Japanese traces exhibit different properties from French traces: they are much larger (>3GB), counts much more packets (> 6 millions) and the upload traffic ratio represents a large part of the overall traffic (> 75%). All the traces count about 4,000 IPs with a peak of IPs at 4,200 IPs for SOP_Fb and SOP_Jx.

For TVAnts, all the traces exhibit roughly the same properties. If we pay more attention, the French traces are slightly larger then the Japanese one, count more packets and present higher upload traffic ratio. It is a different behavior from the other applications where Japanese traces are usually larger than the French one. All the traces count about 1,850 of IPs, whatever their location.

It is interesting to notice that the size of TVUPlayer or SOPCast traces is much smaller than PPSTream or TVAnts one but count much more packets. We investigated this point and observed that –as an example– 80% of the TVUPlayer packets are smaller than 200 Bytes while 2/3 of the PPSTream packets are larger than 1000 Bytes. This should be due to the use of the transport protocol (UDP for TVUPlayer and SOPCast).

Since a peer aims to download the video once but can upload it several times, one can expect that the upload ratio for the larger traces is more important than for the smaller traces. This is the case for TVU_Jy, TVU_Jz, SOP_Jx or SOP_Jy. The larger traces testimony of a more important upload activity than the smaller traces. However, if we compare the amount of downloaded data, it ranges from 512 MB for TVU_Fb to

682 MB for TVU_Jz. It is an important difference since all the traces have the same duration (2h45). This additional amount of downloaded traffic comes from the signaling traffic generated by the remote peers. A peer that uploads a lot of video data to many remote peers will receive in return a large amount of signaling traffic from those peers that orchestrate their download. This observation is at first trivial but, in fact, it can be an important issue in several scenarios. An altruistic peer serving video for many other peers will receive in turn a large amount of signaling traffic. It will waste its bandwidth resource to download the video and could directly affect the video quality if its access link becomes saturated. This is all the more true for TVUPlayer or SOPCast that rely mainly on UDP, which does not have any congestion control mechanism to adapt the throughput of the video streams to the link capacity.

To sum up our general observations of the traces, we observed that PPSTream shows the same behavior for all its traces except for the number of IPs of PPS_Fa and PPS_Jy. The Japanese traces of TVUPlayer and SOPCast are much larger than the French one. SOPCast counts nonetheless the same number of IPs in each trace, and so does TVUPlayer, TVU_Fa excepted. On the opposite, TVAnts shows slightly more traffic for the French traces than the Japanese one, but counts approximately the same number of IPs. SOPCast has clearly two distinct behavior for the French-side and the Japanese-side while the PPSTream's traces are homogeneous. TVants has a slightly different behavior for the French traces and TVUPlayer has a different behavior for all the Japanese one.

4.4.1 Similarity of Peers in the Traces

Table 4.2 shows the total number of IPs that are present in each trace, and the number of IPs labeled as being "IP video". To label an IP, we used the same heuristic as presented on chapter 3. As a brief reminder, an IP is labeled as being an IP video if it exchanges at least 10 packets larger than 1000 Bytes. We consider that an IP address represents a single peer; each IP present in the trace is a peer with which traffic has been exchanged.

The P2P-TV traffic has been collected with multiple measurement points and we wanted to study the similarity of the peers in each trace. The similarity of peers is the ratio of IPs present in the trace with regards to the total number of IPs in all the other traces of the same application. For instance, on the table 4.2, TVA_Fa counts 1,854 IPs. 97% of the TVA_Fa IPs is also present in the other TVAnts traces.

The similarity of peers allows us to study whether our controlled nodes exchanged traffic with the same set of remote peers or not. We believe that it could provide insights on the number of peers in the network. For instance, if all our nodes in France and in Japan have a high ratio of similarity, then all our nodes would have exchanged traffic with the same set of peers. The number of peers in the network should probably be close to the number of peers in the traces.

We computed the IP similarity for the total number of IPs, but also for the IPs video. For the IP video, we observed that some IPs were active only on upload, download or on both sides. Thus, we refine the number of IP video by providing these information.

Table 4.2: Number of peers in each trace.

PPSTream	Total		Video				
	#IP	Similarity (%)	#IP	Similarity (%)	#UP	#Down	#Both
France A	2,625	68	1,060	64	1,047	628	615
France B	3,317	57	1,153	54	1,141	676	664
France C	3,224	59	1,121	57	1,113	655	647
Japan X	3,421	54	1,200	51	1,191	620	611
Japan Y	2,544	67	1,103	59	1,096	586	579
TVUPlayer	Total		Video				
	#IP	Similarity (%)	#IP	Similarity (%)	#UP	#Down	#Both
France A	2,122	46	510	88	483	285	258
France B	1,262	66	478	82	452	267	241
France C	1,093	70	414	88	404	227	217
Japan X	1,111	89	570	85	563	300	293
Japan Y	1,034	90	601	85	599	329	327
Japan Z	1,064	92	565	86	560	325	320
SOPCast	Total		Video				
	#IP	Similarity (%)	#IP	Similarity (%)	#UP	#Down	#Both
France A	3,755	79	845	58	489	418	62
France B	4,268	73	817	52	527	355	65
France C	3,920	76	872	57	543	406	77
France D	3,925	80	877	48	539	422	84
Japan X	4,269	81	818	51	766	262	210
Japan Y	4,048	83	859	52	760	332	233
TVAnts	Total		Video				
	#IP	Similarity (%)	#IP	Similarity (%)	#UP	#Down	#Both
France A	1,854	97	427	87	413	185	171
France B	1,864	97	414	88	382	229	197
France C	1,768	97	429	86	411	211	193
France D	1,887	97	399	88	377	183	161
Japan X	1,855	97	399	86	380	201	182
Japan Y	1,862	97	411	86	392	197	178
Japan Z	1,877	97	414	86	397	201	184

To clarify, an IP active on "both sides" is also counted on the table 4.2 in "download video" and "upload video". From the previous example, TVA_Fa has 427 IPs video; 171 are active on both sides, 185 in download and 413 in upload. It means that 14 IPs are active only in download (185-171) and we find again the 427 IPs video (413+14).

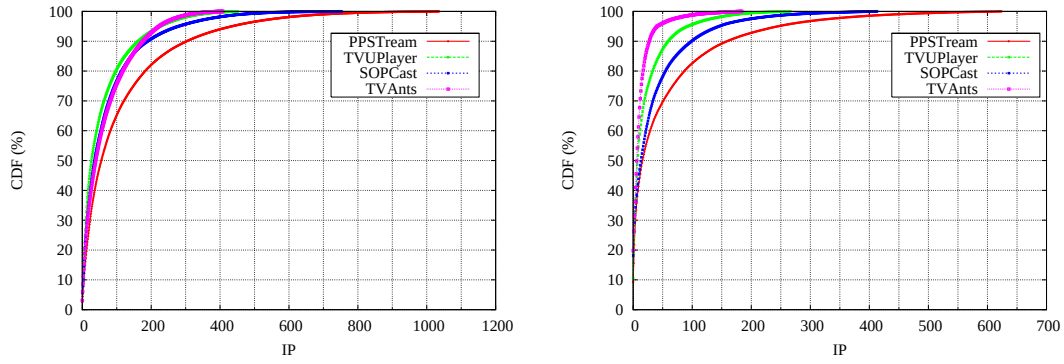
As observed in the previous section, the applications count a very different number of IPs, starting from 1,034 for TVU_Jy to 4,269 for SOP_Jx. Within each application, the number of IPs can be approximately the same for each trace as for TVAnts or SOPCast, It can also present some unexpected behavior as for PPSTream (PPS_Fa and PPS_Jy) or TVUPlayer (TVU_Fa). Regarding the number of IPs video, it is homogeneous within each applications. However, there are still differences between the applications: approximately 400 IPs video for TVAnts, or 1,000 for PPStream.

TVAnts presents a high ratio of similarity (97%) for the overall IPs. When the non IPs video are filtered out, it still exhibits a high ratio of similarity (about 88%). For TVAnts, our controlled peers exchange traffic mostly with the same set of remote peers whatever they are located in France or Japan . PPSTream exhibits also a slight decrease between the overall number of IPs and the IPs video but at a lower rate compared with TVAnts. For the overall IPs, PPStream presents a ratio of similarity at about 60% and it is only about 55% for the IP video. SOPCast exhibits a high ratio of similarity for the overall IPs (about 80%), but suffers an important decrease when the non IPs video have been filtered out (about 53%). TVUPlayer has a different behavior from other applications. For the overall IPs, except TVU_Fa, which counts a large number of IPs compared with other traces, the French traces exhibit a high ratio of similarity (65%) but not as high as the Japanese traces (about 90%). Even thought the video IP similarity decreases a little bit for the Japanese traces, it increases largely for the French traces and all the traces present about 80% of IP video similarity.

A high ratio of similarity from our multiple measurement points may help to infer approximately the total number of peers in the channel For instance, TVAnts counts about 1,850 IPs in each trace and more than 90% of the IPs is present in all the other traces. The collected traces capture probably all the peers watching the same game; there should be roughly 2,000 peers in the network.

For SOPCast, the ratio of similarity is about 80% for each trace which counts about 4,000 IPs. It is a high ratio of similarity and we could assume that our traces capture about 80% of the overall peers in the channel. If we apply the same rationale, the number of peers watching the game with SOPCast would be approximately 20% more, for 5,000 peers.

It is not so trivial to jump to these conclusions for TVUPlayer since there is a huge difference between its IP similarity ratio and IP video similarity ratio. Even though the IP video similarity is high for TVUPlayer, it is not possible to conclude based only on the IP videos since there is a huge difference between the overall IPs and the video IPs. In the same way, PPSTream exhibits a too low ratio of similarity to conclude on the global population of the network. For TVUPlayer and PPStream, the use of additional measurement points would have helped to increase the ratio of similarity in order to deduce the number of peers in network.



(a) Upload (b) Download

Figure 4.2: Traffic by peers (CDF)

4.5 Exchange of Data between Peers

In the previous section we provide general observations on the collected traces. Here, we wanted to observe the amount of traffic exchanged with each remote peer (i.e. IP address). We focus only on video IPs and not on the overall IPs because they are the truly active IPs that exchange video chunks and contribute for almost all the traffic.

On figure 4.2, we plotted the cumulative distribution functions of the traffic exchanged with all the video IPs. For each application, we chose only one trace because it is representative of the other traces of the same application. Figure 4.2(a) is for the upload and figure 4.2(b) for the download. It is worth noticing that the total number of IPs is not the same for the upload or the download side, as shown on table 4.2. On these two figures, the x-axis is for the IP ranking. The IP ranked 0 is the IP with which our controlled peers exchanged the largest amount of traffic. IP ranked 1 is the second contributor, and so on. The top IP on the upload side is not the same IP on the download side. The y-axis cumulates the amount of traffic.

From figure 4.2(a), we observed that PPSTream counts more than 1,000 IP video in upload, 650 for SOPCast and no more than 500 for the two remaining applications. Even though these IPs have been labeled as being video (section 3.3), only 30% of the IPs video contributes for more than 90% of the upload traffic. From figure 4.2(b), the number of download IPs is different for all the applications and range from about 180 for TVAnts to 620 IPs for PPStream. The top contributor on the upload and the download side are presented on table 4.3. It ranges from 2.6% for PPSTream to 5.7% for TVUPlayer in upload. Since there is less IPs active in download, the top download IP contributes at a higher rate: from 9.2% to 19.8% TVAnts. Less than 25% of the video IP are sufficient to generate 90% of the downloaded traffic.

Besides the amount of traffic exchanged in upload and download by each peer, we are interested to observe the “sharing ratio” between peers. The sharing ratio is the amount of traffic a peer uploads divided by the amount of traffic it downloads. The

Table 4.3: Amount of traffic for the top three IPs

	Upload (%)			Download (%)		
	IP 1	IP 2	IP 3	IP 1	IP 2	IP 3
PPSTream	2.6	1.8	1.8	9.2	6.4	6.1
TVUPlayer	5.7	2.9	2.6	10.5	8.2	7.1
SOPCast	3.1	2.5	2.3	18.2	8.1	2.9
TVAnts	3.1	3.0	2.7	19.8	6.6	5.2

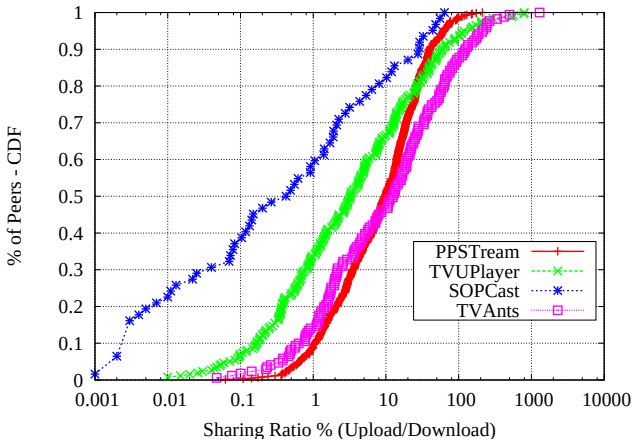
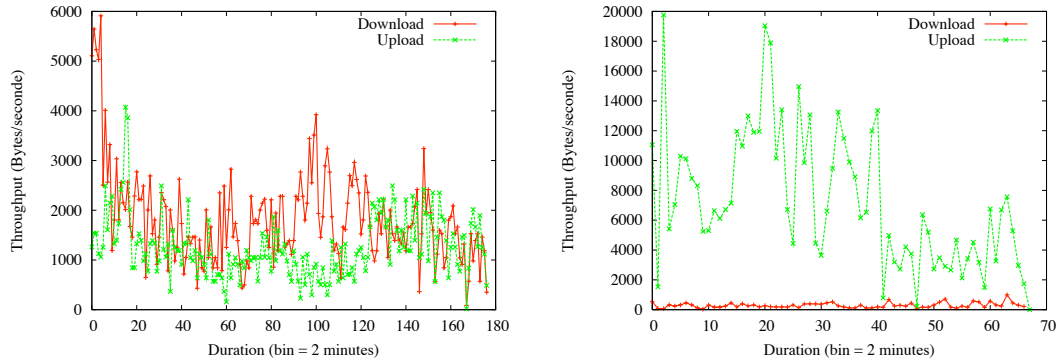


Figure 4.3: Sharing ratio between peers (upload/download).

sharing ratio allows to connect the upload and the download traffic of a peer. Indeed, the top contributor peer on one side does not necessarily contribute as much as the other side. For all the applications, the figure 4.3 shows the cumulative distribution function (CDF) of the relationship between the sharing ratio (upload/download) between a peer which we control, and all the peers with which it exchanges data. Again, we show a single trace for each application because it is representative for the other traces. When the ratio is above 1, our peer is altruistic towards another peer; when equal to 1, the exchange is fair; when below one, our peer is taking advantage of the altruism of others. For PPStream and TVAnts, 10% of peers are altruistic with our peer and transmitted more data than they received. The 90% remaining are beneficiaries of our altruism. This behavior is the same for TVUPlayer at distinct rate: 30% of peers are altruistic and the remaining 70% benefits from our altruism. This is the opposite for SOPCast: the majority of the remote peers transmitted more data than they received (60%).

Overall, the traffic exchanges for all the applications are never fair to our controlled peers, with the large majority being mainly consumers (ratio>1) than producers (ratio<1). This is still an unfair sharing ratio for SOPCast but it is the opposite: a large majority of remote peers are altruistic with our peer.



(a) Fair exchange between peers

(b) Unfair exchange between peers

Figure 4.4: Traffic exchanges between one of our controlled node and two remote peers.

However, there exists distinct peer behavior traits as those presented on the figure 4.4. Looking at figures 4.4(a) and 4.4(b) we have isolated uplink and downlink traffic, between one of our peer and others. In the figure 4.4(a), the two curves are not the same, but show that two peers try to contribute in the same proportion during their exchange. In contrast, in figure 4.4(b) we can see that our peer has contributed a lot of information while it has received virtually nothing in exchange. Clearly, whatever the applications or controlled peers, the second case (figure 4.4(b)) is the case that is the most likely to happen; unfair exchange between peers without any reciprocity.

These results shows that fairness is not achieved in P2P live streaming systems. Since each peer is in charge to forward the traffic to other peers, each peer should forward the same amount of data it received or at least contribute with its available resources. If it is not the case, the peers are not helping the other peers by sharing their resources in the network and this is a major issue for the scalability of these P2P systems.

Remember these applications claim to use mesh-based architecture inspired by BitTorrent. BitTorrent has been successful because it relies mainly on the use of its tit-for-tat incentive mechanism. These observations illustrate the need for an incentive mechanism which is designed for P2P systems for distribution of live multimedia flows. Such a mechanism must allow all the peers truly to collaborate in the network in order to achieve fairness and scalability in the network.

4.6 Distance between Peers

In the previous section, we discussed about the amount of traffic exchanged with each peer and also introduced the sharing ratio metric. The study of this metric shows that there is no fairness with the measured P2P-TV systems because the data exchanges were not reciprocal. In this section, we study the distance between peers and the locality

Table 4.4: Default TTL value for common operating systems.

Operating System	default TTL
FreeBSD 2.1R	64
Linux 2.6.30	64
Mac OS X 10.5.6	64
Solaris 2.x	255
Windows 95	32
Windows 98/2000	128
Windows XP/Vista	128

of the traffic. In other words, we wanted to examine the distance where our peers fetch the video traffic from the remote peers on the Internet. Then, in the following, we will study the distance in number of hops between peers. We will also study the exchange of traffic at the AS-level and the country-level.

4.6.1 Number of hops

In order to infer the distance between peers, we investigate the IP datagram of the packets. The IP header has a TTL field, which is decreased at each hops by the routers of the Internet. The various operating systems use a different default TTL for the datagram they send. The default TTL for common operating systems are presented on table 4.4.

Given that the default value used for the TTL is usually a multiple of two the chances are that, if a packet arrives with a TTL of 120 the packet was originally sent with a TTL of 128 [112]. In the same way if a packet is received with a TTL of 60, chances are that the original packet had a TTL of 64, unless the originating system has been explicitly tuned to use a non-default value. In our opinion, we believe that regular Internet users do not change the default value of their operating system. Only skill-computing users would possibly change this value to their needs. Moreover, this discussion assumes there was no protocol scrubber, transparent proxy, or some other middle-box that overwrites the TTL field in a non-standard way, between the originating system and the point of the network in which the packet was received (i.e. our measurement points).

In the case of our study, the P2P-TV applications are built for Windows XP, which sets the default TTL to 128; we were therefore expecting that the IP datagrams have been sent with this default TTL. Thus, the TTL field of each received IP datagram reveals the distance in number of hops from the source to its destination. We then compute this distance H as follows:

$$H = defaultTTL - receivedTTL$$

We used this heuristic to compute the distance in number of hops from our nodes to

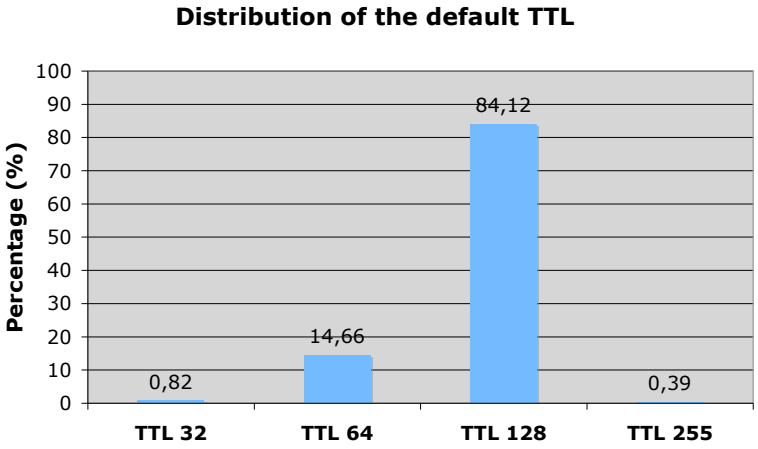


Figure 4.5: Distribution of the initial TTL value of the IP datagrams.

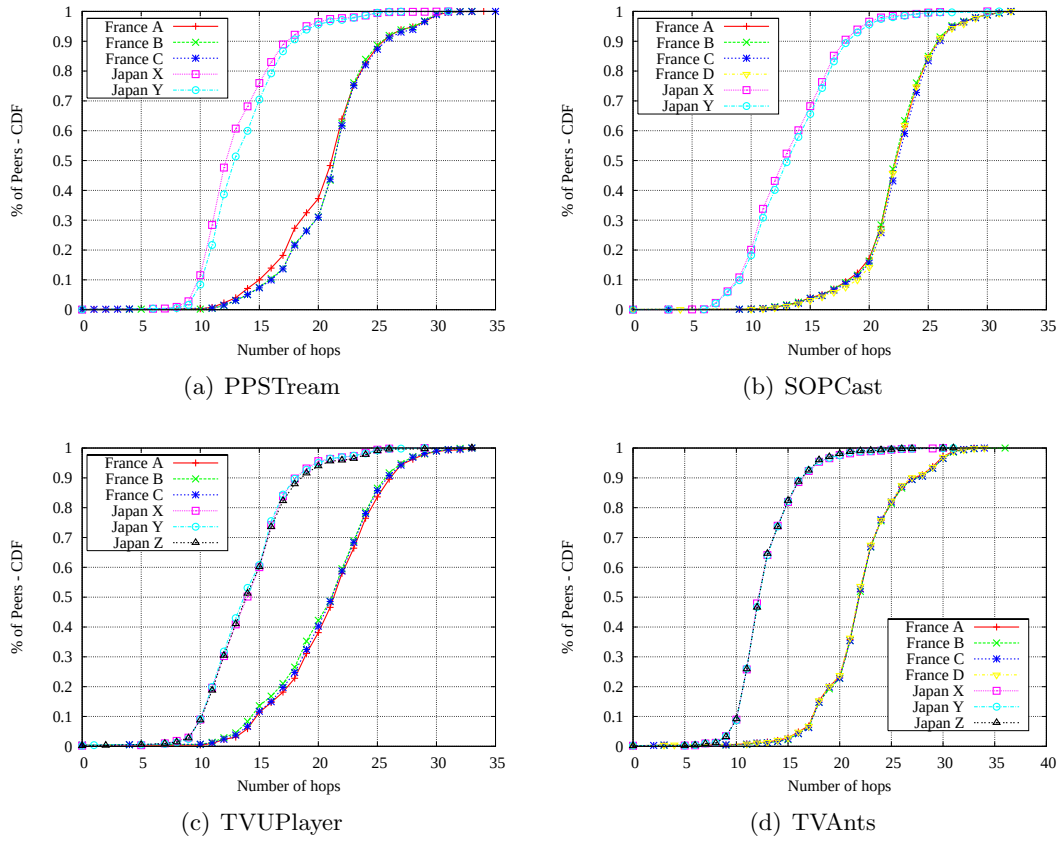
the remote peers with which our nodes exchanged traffic. This heuristic is practicable for the received IP datagrams and not for the sent datagrams. Indeed, in upload, our datagrams are sent with the default TTL value (128), and we could not inspect the TTL field of these datagrams when they reached their destination.

In order to verify the accuracy of our heuristic, for each received IP datagram, we compute H with all the most common default TTL values: 32, 64, 128 and 255. By comparing the different results provided with the heuristic, the initial default TTL is the one that gives the smallest positive value. For instance, if the received TTL is 107:

$$\begin{aligned} H_{32} &= 32 - 107 = -75 \\ H_{64} &= 64 - 107 = -43 \\ H_{128} &= 128 - 107 = 21 \\ H_{255} &= 255 - 107 = 148 \end{aligned}$$

The smallest positive value is 21, obtained with H_{128}; the initial default TTL of the IP datagram was 128.

On figure 4.5, we plot the distribution of the initial default TTL for all the datagrams we received in all the presented traces. Clearly, the large majority of the default TTL is 128 (84.12%). It was expected since the these applications are built for Windows XP, which uses 128 as a default TTL value. An important part of the default TTL was set to 64 (14.66%). We believe that Network Address Translator (NAT) has rewritten the TTL field of these translated packets such that this field is set homogeneously among all packets forwarded toward the external network [113]. A very few number of default



(a) PPSTream (b) SOPCast

(c) TVUPlayer (d) TVAnts

Figure 4.6: Distance in number of hops between our controlled nodes and the remote peers on the Internet.

TTL was set to 32 and 255 (0.82% and 0.39% respectively). We assume that it is also due to other kind of NAT. The number of such packets is very limited and will not affect the results.

Since we are confident with our heuristic to infer the distance –in number of hops– between our controlled peers and the remote peers on the Internet, for each trace, we plot on figure 4.6 the cumulative distribution function of this distance. Each subfigure is for the different applications. On the x-axis, we show the number of hops from the remote peers to our nodes. The y-axis shows the percentage of peers.

All the applications show two distinct behavior for the Japanese traces and the French traces. For our French nodes, 50% of the remote peers are situated at a distance of more than 20 hops. For the Japanese nodes, it is only about 12 hops for 50% of the remote peers. Moreover, 90% of the peers are situated at less than 20 hops of our Japanese nodes. For the French nodes, 90% of the peers are situated at less than 25 hops.

Clearly, for all the applications, our French peers download the video at further distance than the Japanese peers.

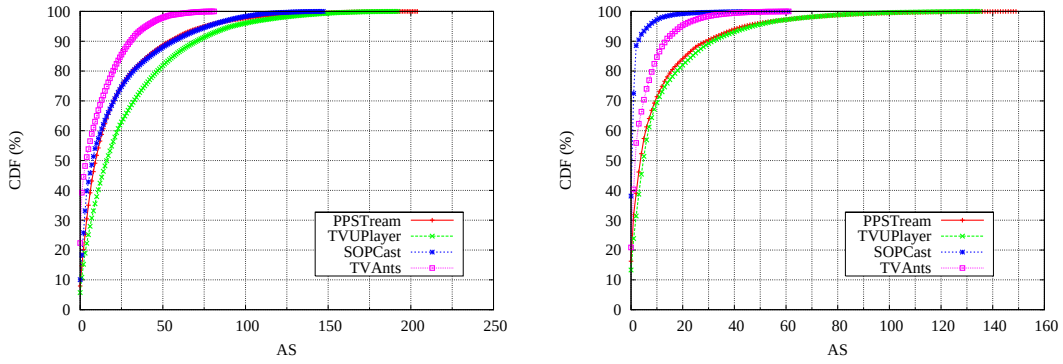
Several assumptions could explain this trend. First, P2P-TV applications are already very popular in Asia and more precisely in China: most of the peers are probably situated in Asia. The use of P2P-TV in Europe is still limited and even PPSTream has the same behavior as the other applications whereas it was broadcasting a program of interest for European users (UEFA Champions league). If the large majority of peers are situated in Asia, then the French nodes fetch the content from peers at further distance compared with the Japanese nodes. Second, since commercial applications have been released from China, one could expect that most of the source servers of the video are situated in Asia. Thus, our peers aim to fetch the content from these source servers far from them, as it is also the case for the remote peers. Third, we could also imagine that these applications do not use any locality-aware mechanisms to select the provider peers. Otherwise, the French nodes would have downloaded the video at a closer distance. If the applications were using any locality-aware mechanism, a possible scenario could be that one of our French nodes downloads the video far from it and forwards the content to the other French nodes situated in the same network. In this case, we would observe on the figure 4.6 only a single French peer that downloads far from it and the other get the data at a closer distance. At the end, the long distance for the French peers to get the video could be an issue since it may increase the end-to-end delay to receive the video packets and affect the quality of the video.

4.6.2 Autonomous Systems and Countries

To extend our study of the distance between peers and the locality of the traffic, it seemed necessary to study the traffic at a higher granularity. We therefore investigate the amount of traffic exchanged with the different autonomous systems (ASes) and countries. An autonomous system is a group of IP networks under a single technical administration (i.e. ISP) which has a clearly defined routing policy [114]. Even though the network of ISPs can be divided into several ASes, the number of ASes involved in the exchanged could provide some insights on the traffic exchange between ISPs. The country granularity –i.e. the geographic location of peers– would provide information on the behavior of users and the worldwide P2P-TV usage.

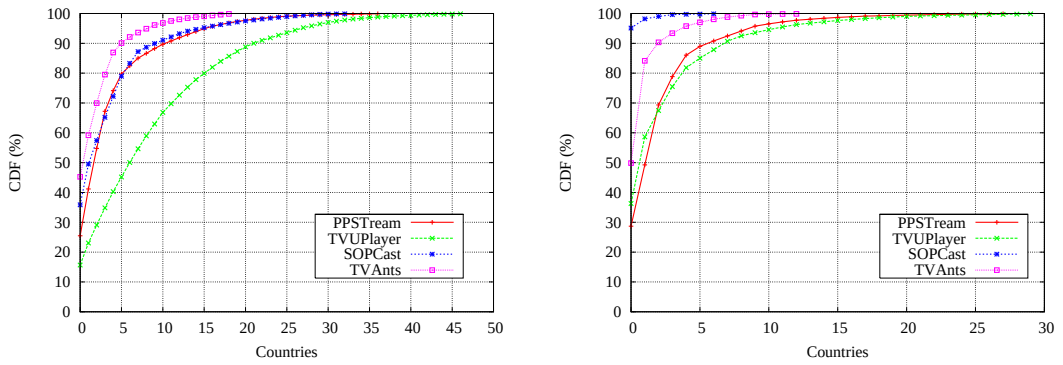
To this end, we mapped each video IP address to its parent AS and origin country by using the `whois` [115] service provided by cymru [116]. This `whois` service add some functionalities to the standard one; it supports bulk IP submissions when combined with GNU netcat [117] to optimize the queries. We mapped a peer's IP address to its country by querying the free MaxMind GeoIP database [118]. We still consider only IPs video because they are the only one to generate traffic.

On figure 4.7, we plot the cumulative distribution function of the volume of traffic exchanged with each AS, and on figure 4.8 with each country. There are two subfigures for each level of granularity: one for the upload traffic and one for the download. On the x-axis, the entities (ASes or countries) are sorted by their volume of traffic. The AS 0 on the x-axis is the top AS with which one of our peer exchanges the largest amount



(a) ASes Upload (b) ASes Download

Figure 4.7: Traffic by ASes (CDF)



(a) Countries Upload (b) Countries Download

Figure 4.8: Traffic by countries (CDF)

of traffic. AS 1 is the second one. The countries are ranked in a similar manner. To avoid any ambiguity , we insist on the fact that the x-axis is the AS ranking; it is not related to the *AS id*, which is used to identify each AS on the Internet. The y-axis cumulates the amount of traffic. Each side of the traffic is studied independently from the other: the top AS (i.e. country) on the upload side is not the same AS (i.e. country) on the download side. For each application, we show only one trace because it is representative for the other traces of the same application. The top three ASes and countries (upload and download) are also presented on the tables 4.5 and 4.6.

Whatever the traffic direction, hundreds of ASes are implied in the traffic exchanges. The upload traffic involves much more ASes than the download traffic. Applications such PPStream involves more than 200 ASes in upload: twice as much ASes as TVAnts. For PPStream and SOPCast, 60 ASes count for 90% of the upload traffic, but PP-

Table 4.5: Amount of Traffic for the top three Ases

	Upload (%)			Download (%)		
	AS 1	AS 2	AS 3	AS 1	AS 2	AS 3
PPSTream	7.9	6.8	5.5	16.3	15.6	7.2
TVUPlayer	5.7	5.1	4.4	13.3	10.5	7.6
SOPCast	10.1	8.3	7.4	38.1	34.4	16.0
TVAnts	22.4	16.9	5.3	20.9	19.5	15.6

Table 4.6: Amount of Traffic for the top three Countries

	Upload (%)			Download (%)		
	Co. 1	Co. 2	Co. 3	Co. 1	Co. 2	Co. 3
PPSTream	25.5	15.7	13.6	28.7	20.5	20.1
TVUPlayer	15.7	7.4	6.0	36.3	22.4	8.8
SOPCast	35.9	13.6	7.9	95.1	3.1	0.8
TVAnts	45.2	13.9	10.8	49.9	34.3	6.2

STream exchange traffic with much more ASes at the end (150 for SOPCast). TVUPlayer uploads traffic to almost the same number of ASes as PPStream but it is more balanced between each AS: 90% of its traffic goes to more than 75 different ASes. TVAnts uploads traffic to a reduced set of ASes compared to the other. Only 30 ASes count for 90% of its upload traffic: half as many ASes as the other applications. For all the applications, the download traffic comes from a smaller number of ASes than the upload traffic. SOPCast downloads 90% of its traffic from only 3 ASes. TVAnts downloads from about 10 ASes while TVUPlayer and PPSTream needs almost 30 ASes.

By comparing the upload and download traffic, the behavior of TVUPlayer and PPSTream look the same whatever the traffic direction, even though they use less ASes for the download. TVAnts uses almost the same number of ASes in upload and download. SOPCast presents the more surprising behavior: it is similar to the other applications on the upload side; but it downloads the content from only a very few ASes.

At the country level, TVAnts still shows the same behavior in upload and download and involves approximately the same number of countries on both sides. SOPCast downloads the traffic from only a very few number of countries. TVUPlayer and PPSTream count many countries in their traffic. In upload, TVUPlayer is the only application to involve as much countries since 90% of its traffic reaches about 20 countries while it is only 10 countries for PPSTream and SOPCast or 5 countries for TVAnts.

In order to obtain the number of AS hops between our controlled nodes and the other peers on the Internet, we proceed as follows. For some of our experiments, we

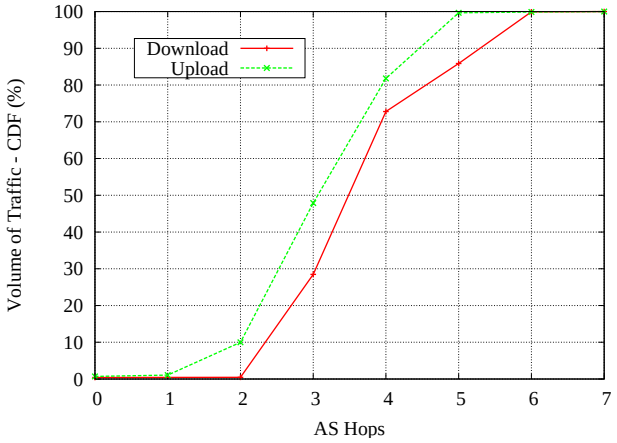


Figure 4.9: Distance in number of AS hops between our controlled nodes and the remote peers on the Internet.

performed a `traceroute` towards each destination. The `traceroute` is a common tool for network diagnosis and measurement tool. It allows discovering the route packets take to network host. Since our nodes were running Windows XP, we used traceroute with Cygwin [119], which is an Unix-like environment and command-line interface for Microsoft Windows. It is worth to underline that, instead of the common traceroute, we used paris-traceroute [120, 121] which is an implementation from the Université Pierre et Marie Curie that shows its efficiency to discover more accurately the route to hosts [122].

Again, we obtained the corresponding AS of each network interface of a route by using the whois service provided by cymru. By aggregating the similar consecutive AS of a route, it was therefore possible to deduce the number of AS hops between our nodes and each destination. At the time of our experiment, for practical reasons, the traceroutes have been performed only from our nodes situated in France. When we analyzed the traceroutes, many end-destinations were unreachable. Thus, the results are not very accurate and are only an approximation of the distance in number of ASes from our French controlled peers to the remote peers. We chose to keep this result in our study because it provides some insights on the number of AS hops to get the video but it has to be considered carefully.

On the figure 4.9, we show the cumulative distribution function of the traffic according to the number of AS hops. The x-axis shows the number of AS hops while the y-axis cumulates the amount of traffic. Since these results are not completely accurate, we present only the results for SOPCast but the other applications show similar results.

On the figure, we observed that 50% of the upload traffic is sent to 3 AS hops or less, whereas only 30% of the download traffic come from this distance in number of AS hops. 15% of the traffic is also downloaded at 5 to 7 AS hops while the upload traffic never goes beyond 5 AS hops.

For our nodes situated in France, the download traffic has transited among much more ASes compared with the upload traffic. Even if we observed that there is a larger number of ASes involved in the upload traffic than the download traffic, the downloaded traffic comes from a further distance compared with the upload traffic.

We noticed previously that the nodes situated in France download the video at farther distance than the Japanese one. Moreover, for the French nodes, the download traffic has transited into much more ASes compared with the upload traffic. Regarding the traffic at the AS-level or the country-level, a large number of ASes or countries are involved in the exchange and it is especially the case in the upload side.

In this section, we wanted to point out the fact that P2P traffic is a main concern for ISPs. The P2P traffic imposes significant traffic engineering challenges to ISPs. By selecting peers arbitrary to exchange the traffic, the fact is that P2P systems reinvent and re-implement a routing system whose dynamics breaks the effort made by ISPs to perform an efficient traffic engineering in their network to optimize their routing. For ISPs, it is important to keep the P2P traffic local. To this end, there have been some proposals that suggest that P2P network should build the overlay topology by using locality-aware mechanisms and choosing closer peers [123–126]. In our case, the P2P-TV traffic is exchanged with many ASes of different ISPs. It may overload the links between ISPs and it is potentially costly for the ISPs. All these results emphasize that P2P-TV systems do not pay attention to the locality of peers when exchanging traffic. This is a major issue for the ISPs and the traffic engineering they perform to provision the resources of their network and predict its load.

4.7 Geographic Location of Peers

In this section, we study the geographic location of peers and the volume of traffic exchanged by countries. Based on our mapping of IP with their corresponding countries, we plot for each trace the stacked histograms of the volume of traffic by countries, and the number of peers by countries on the figures 4.10 to 4.13. Each application is represented by four figures: two are for the volume of traffic exchanged by countries and the two remaining figures are for the population by countries (i.e. number of peers). Again, we separated the upload and download traffic in our study. On each figure, every trace is represented by its own histogram.

For all these plots, the legends are similar and present about 20 countries. These countries were the most represented countries in our traces. However, remember on figure 4.8, we showed that some traces count about 40 different countries. Some countries were not present in all the traces and did not contribute for a large part of the traffic. When the amount of traffic does not exceed 1% for a country and this country is present in only a few traces, we do not show this country on the legend. Instead, such country is included under the label “other” of the legend, which gathers all the small contributor countries. In this study, we took into account only the IPs video which are truly exchanging traffic.

As an example, if we consider the figure 4.10 for PPSTream and the trace PPS_Fb:

on figure 4.11(a) the ratio of upload peers are distributed as follows: 35% of peers are from China, 15% from Hong Kong, 5% from France, 15% from United Kingdom (UK), 10% from a set of countries labeled as "other", and the rest comes from other countries of the legend. From the table 4.2, the exact number of peers by countries can be deduced (PPS_Fb counts 1,141 upload peers). On figure 4.11(b), 15% of the upload traffic goes to China, 5% for Honk Kong, 25% for UK, 15% for France, about 10% for Spain, 10% for "other" countries and so on.

4.7.1 Influence of the Content on the Population

As stated in section 4.3.3, we remind that all the measured events were soccer games. For PPSTream, the game was the preliminary round of the champions' league between Toulouse and Liverpool. The other games were qualifying rounds for the Olympic tournament: Japan vs. Vietnam with TVUPlayer, China vs. Japan for SOPCast and China vs. Vietnam for TVAnts.

As for traditional TV, we were expecting that the population watching the games depends directly on the broadcasted content. In other word, we were expecting to observe a large amount of peers from Japan for the games involving Japan or a large number of peers from UK for the games with Liverpool. Clearly, this is a basic statement and the sociological usage of these systems should be deeper investigated. The broadcasted soccer games involved the following countries in particular: United Kingdom, France, China, Japan and Vietnam. Moreover, France and Japan are also the location of the measurement experiment. We will pay a particular attention on the presence of these countries in our traces to investigate the effect of the broadcasted content on the P2P-TV population.

For PPSTream (figure 4.10), it was expected to notice peers from UK and France because the soccer games was opposing teams from these two countries. In our PPSTream traces, peers from UK represent about 10% of the total number of peers and constitute a significant part of the volume of upload traffic. Differently, the peers from France represent roughly 5% of the peers but, in comparison, it has a more important volume of traffic, especially in the download side. However, we observed also a significant part of peers from UK with SOPCast (figure 4.12). This behavior was unexpected since the game was between China and Japan. The number of UK peers and the related amount of traffic are more important in the French traces of SOPCast than the Japanese ones and contribute for a large part of the upload traffic.

When the broadcasted soccer games involved the Chinese or the Japanese teams, these two countries represent a high ratio of peers and volume of traffic. This is clearly the case for China with SOPCast and TVAnts (figures 4.12 and 4.13), and for Japan with SOPCast and TVUPlayer (figures 4.12 and 4.11). The presence of these two countries is less important during the two remaining games with the two other applications.

Vietnamese peers count for an important part of the upload traffic of TVUPlayer (figure 4.11). We were not expecting to count as much Vietnamese peers in our traces. We thought the internet access conditions for this emerging country were still too bad to allow Vietnamese users to use high bandwidth consuming applications such P2P-

TV [127]. However, we did not observe any Vietnamese peers in the other game with TVAnts (figure 4.13).

As it was expected, the broadcasted contents have an effect on the population of peers and their geographic location. For instance, we observed a large number of peers located in UK with PPSTream; there are also many peers from China or Japan for SOPCast, TVAnts and TVUPlayer, when these two countries are involved in the games. Moreover, even if Vietnam was no present in TVAnts, it was surprising to observe so many Vietnamese peers in the TVUPlayer, especially with regards to the Vietnam Internet access conditions. To mitigate our assumptions, it is worth to highlight that the number of French peers is not as important as the number of UK peers with PPSTream. Moreover, French peers are present in a small proportion in almost every traces of each application even though the content did not involve this country.

This can be due to the fact that a part of our testbed is located in France: it might slightly bias our measurement experiment. However, we did not observe the same behavior with Japanese peers although the second part of our testbed is situated in this country. For Japanese peers, there is a sharp contrast depending on whether the broadcasted content involved Japan or not. This is not obvious to observe with French peers, for which their part is roughly the same whatever the application.

Finally, it is interesting to note that, among all the traces, we did not count a large number of peers from the United States (US). This is the opposite for China which counts a large number of peers in almost every trace. The lack of US users is probably due to several reasons such as the unpopular nature of the broadcasted content (soccer). The broadcasted contents were also scheduled in the morning or during the working time in US. This could prevent potential viewers from using these applications.

Moreover, the high-speed Internet access in US is not largely deployed for users; it could also explain the limited use of P2P-TV applications in US. For instance, Zattoo [98] already launched their application in Europe but it is still not available in US. The reasons are both political (copyright) and technical (Internet broadband access). Differently, P2P-TV applications have been launched by Chinese companies and are already very popular in China. Thus, we were not surprised to count so many Chinese peers in the traces.

4.7.2 Number of Peers vs. Volume of Traffic

Regarding the peers and their locations with PPSTream, all the traces present approximately the same population of peers, whatever their location. For instance, all the traces count roughly 30% of Chinese peers and 20% of peers from Hong Kong. As explained before, the large part of peers from UK can be explained by the nature of the broadcasted content. Some countries represent a totally different part of the population and traffic. As an example, French peers count only about 5% of all the peers but generate a larger part of the volume of traffic: 15% of the upload traffic for the French traces and 5% to 10% for the Japanese ones. In the same way, China counts

more peers than Hong Kong in download (30% and 20%) but generate a less important part of the download traffic.

For SOPCast, we can distinguish distinct traits for the French nodes and Japanese nodes. Our French nodes count almost entirely Chinese peers in download, while the Japanese nodes count a large part of peers from Japan. This is consistent with our previous observation of SOPCast where only one country was able to transmit 90% of the video (table 4.6). Regarding the upload side, if we except the fact that our Japanese nodes still count a large number of Japanese peers while our French nodes transmit to Chinese peers, all the traces count from 15% to 30% of peers located in Europe. However, in term of volume of upload traffic, European peers (UK, Germany and France) constitute a large part of the traffic for our French nodes, even though the majority of their peers are located in Asia. This is not the case for our Japanese nodes where a large number of Japanese peers contribute for the majority of the traffic.

The population of peers in all our TVUPlayer traces is roughly the same on both directions. However, the volume of traffic is not proportional with the population of peers. This is the case for Japan and Korea: they constitute about 10% of peers in download and contribute for 20% or 30% of the traffic. In upload, France, Vietnam or Japan constitute a significant part of the traffic but the population of peers from these countries did not represent a large number of peers.

For TVAnts, the population is similar in upload and download and come from Asia in its vast majority. However, there is no relationship between the number of peers and the volume of traffic. In download, there is only a small part of French peers but it represents a significant part of the traffic in the French traces. This is similar for our Japanese traces which show a large part of traffic from Japan but count only a few peers from Japan. In the upload side, half of the peers are from China and contribute for a large part of the traffic; French nodes upload more traffic towards Europe than Japanese nodes.

At the end, all the traces of TVAnts and TVUPlayer show roughly the same population of peers in upload and download whatever the location of our controlled nodes. However, there is not necessarily a relationship between the population of peers and the volume of traffic. For instance, this is the case for Korea or Vietnam for TVUplayer, France or Japan for TVAnts. With SOPCast, we can distinguish two distinct traits according to the location of our controlled nodes. For PPStream, except the case of UK peers, the population of peers and their traffic is roughly similar for all the traces.

4.7.3 Traffic in Upload and Download

The study of the traffic at the country granularity helped us to observe the traffic exchanges because it aggregates large amount of the traffic. Thus, we can focus on larger part of traffic that is easier to analyze compared with smaller traffic granularities such as AS or IP level (peers)

If we pay more attention to the volume of traffic in both directions, we observed that PPSTream downloads mainly from Asia (China and Hong Kong) while it uploads more in Europe (France and UK). In the same way, SOPCast downloads from China and

Japan; Our French nodes upload mainly to European peers while Japanese nodes upload to Japanese peers. Again, TVUPlayer seems to download mainly from Asia (Korea and Japan) and upload to Europe for the French nodes and Asia for the Japanese nodes (Japan and Vietnam). Whatever our controlled nodes, most part of the traffic was downloaded from China and Honk Kong with TVAnts. Since our nodes upload most part of the traffic to China, French nodes upload also largely in Europe while Japanese nodes upload in Asia. Overall, for all the applications, our nodes download the largest part of traffic from Asia. Then, the nodes located in France upload more to Europe and those in Japan to Asia.

From these observations, we argue that our nodes do not clearly reciprocate when downloading the traffic. For instance, French peers download traffic from Asia but upload in Europe. We can extend this observation to the IP-level (peers). If our peers get the data from a country and forward it to another one, then it do not reciprocate fairly with its provider peers: the exchange of traffic towards different countries involves exchange of traffic towards different peers.

In section 4.5, we already had the same statement when investigating the sharing ratio of peers, and we observed that fairness was not achieved in the P2P-TV systems because peers did not exchange the same amount of traffic between them. The reasons why peers do not exchange the same amount of traffic could be explained with this result.

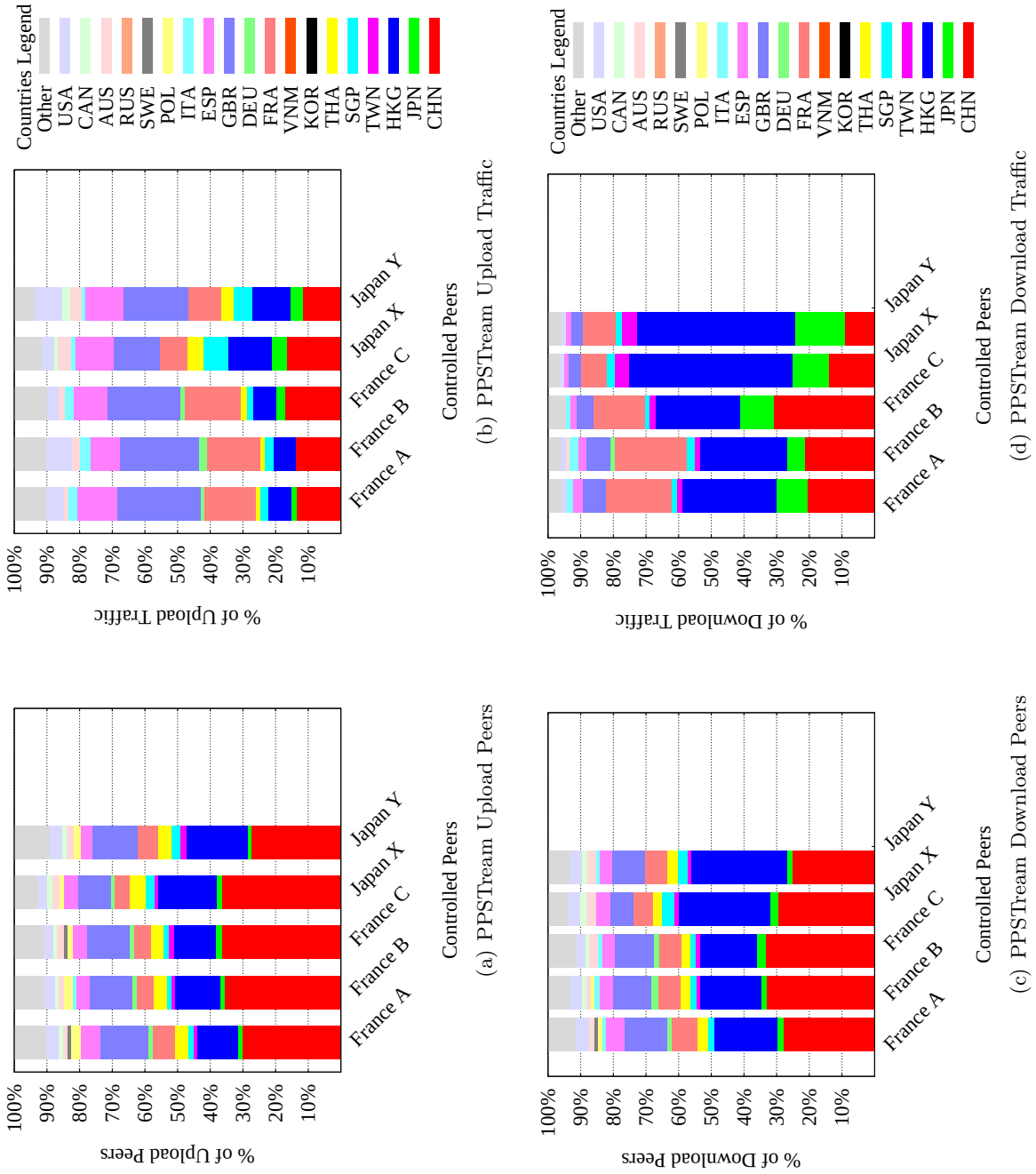
In fact, it seems that the video streams follow a directed path among peers and it is not possible to reciprocate with data to the provider peers. In our example, our French nodes are not able to reciprocate with data to the Asian peers because they are situated above in the play-out point. Thus, they are late in the video playback and do not have any data of interest to transmit in return to Asian peers (ahead in the playback time). They can only transmit to other European peers that are in the same playback time of the video.

This phenomenon could directly come from the long end-to-end network delay and number of network hops from Asia to Europe since we observed previously that our French nodes download the traffic at a further distance than the Japanese nodes (section 4.6). It is also consistent with the fact that the downloaded traffic come from a larger distance in number of AS hops than the upload traffic that reaches closer peers. At the end, it prevents the French peers to reciprocate with data to Asian peers because they are situated in different point of the video playback. For our Japanese nodes, they are situated in similar play-out points as the other Asian peers and can reciprocate with data.

To discuss this finding, one could argue that the non-reciprocal exchange of traffic could come from the asymmetric Internet connection of most of the Internet users with residential connections (DSL). Indeed, with residential access to the Internet, the users have much less upload capabilities than download capabilities (from 2 to 8 times less). However, in the point of view of our controlled nodes, this argument is acceptable only for the received traffic from other peers that have possibly limited uplink bandwidth but not for the traffic sent by our nodes. Indeed, our controlled nodes do have a symmetric access and could upload in the same proportion as they download. In a case

where our peer uploads an amount of traffic to a remote peer which forwards in turn a smaller amount of traffic, one can assume that this is due to the asymmetric Internet connection of the remote peer. For instance, with SOPCast (figure 4.12 in SOP_Jx and SOP_Jy, our Japanese nodes upload a lot of data towards Japanese peers (60% and 40% of upload) and receive only 30% of the traffic in download. Otherwise, our peers upload more than they download and the traffic exchanges are not fair. On the opposite, our nodes downloading a certain amount of data from a remote peer are able to transmit as much data in return because they do have high-speed symmetric access to the Internet. In our traces, this is clearly not the case with SOPCast: our French peers download a lot of traffic from China but do not upload in the same proportion to China. This behavior is similar for PPSTream and TVAnts with Hong Kong, or for TVUPlayer with Korea.

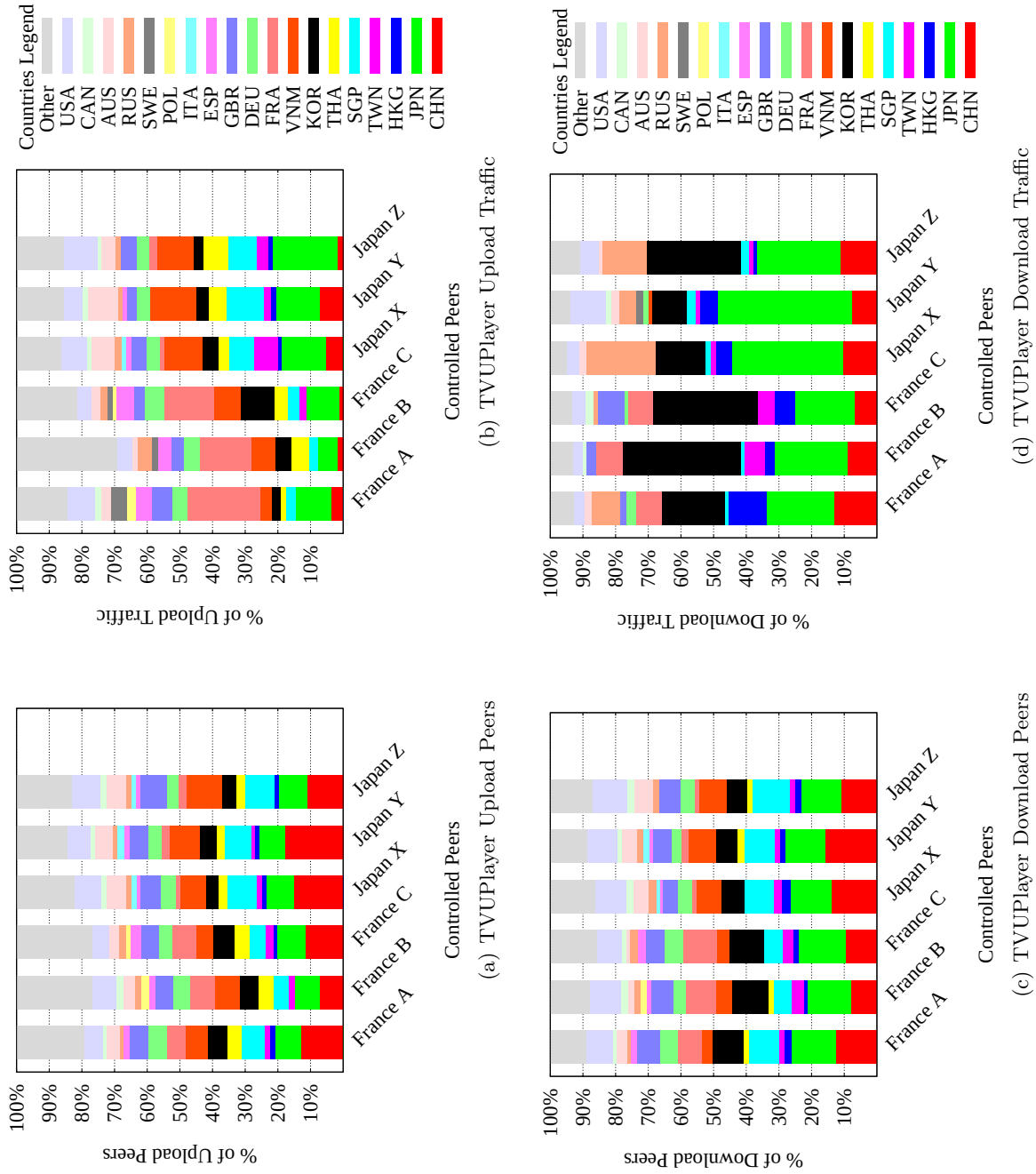
Since our nodes have high and symmetric capabilities but did not upload the same amount of data to the countries from which they download, we believe that asymmetric Internet access is not the only reason that prevents the reciprocal exchange between peers. In fact, multimedia flows impose temporal constraints that result from the continuous nature of the transmission. Peers cannot transmit data in return, not necessarily because they are uncooperative, but because the nature of the content and the temporal constraints make it pointless.



(a) PPSTream Upload Peers (b) PPSTream Upload Traffic

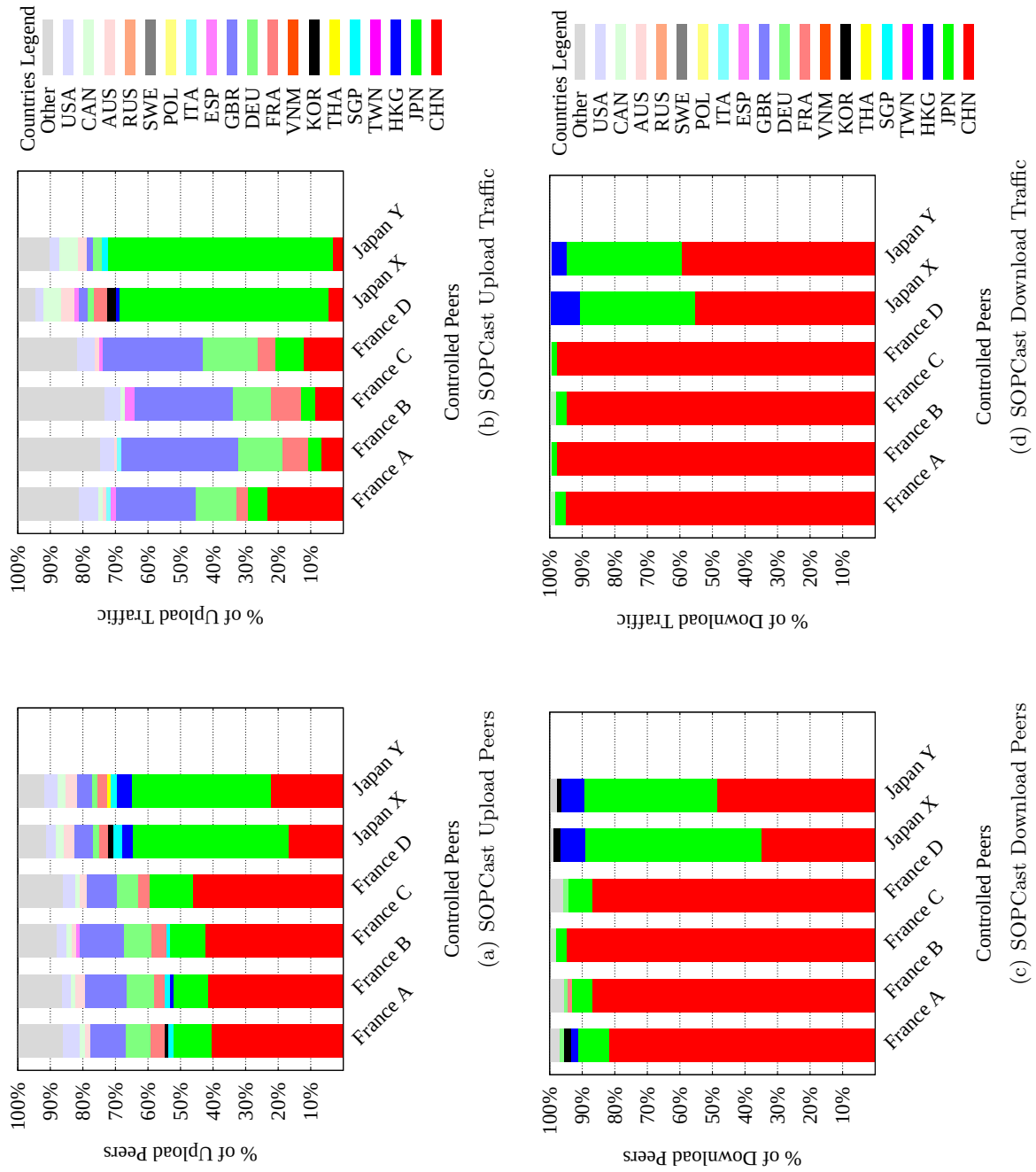
(c) PPSTream Download Peers (d) PPSTream Download Traffic

Figure 4.10: PPSTream: geographic location of peers and traffic



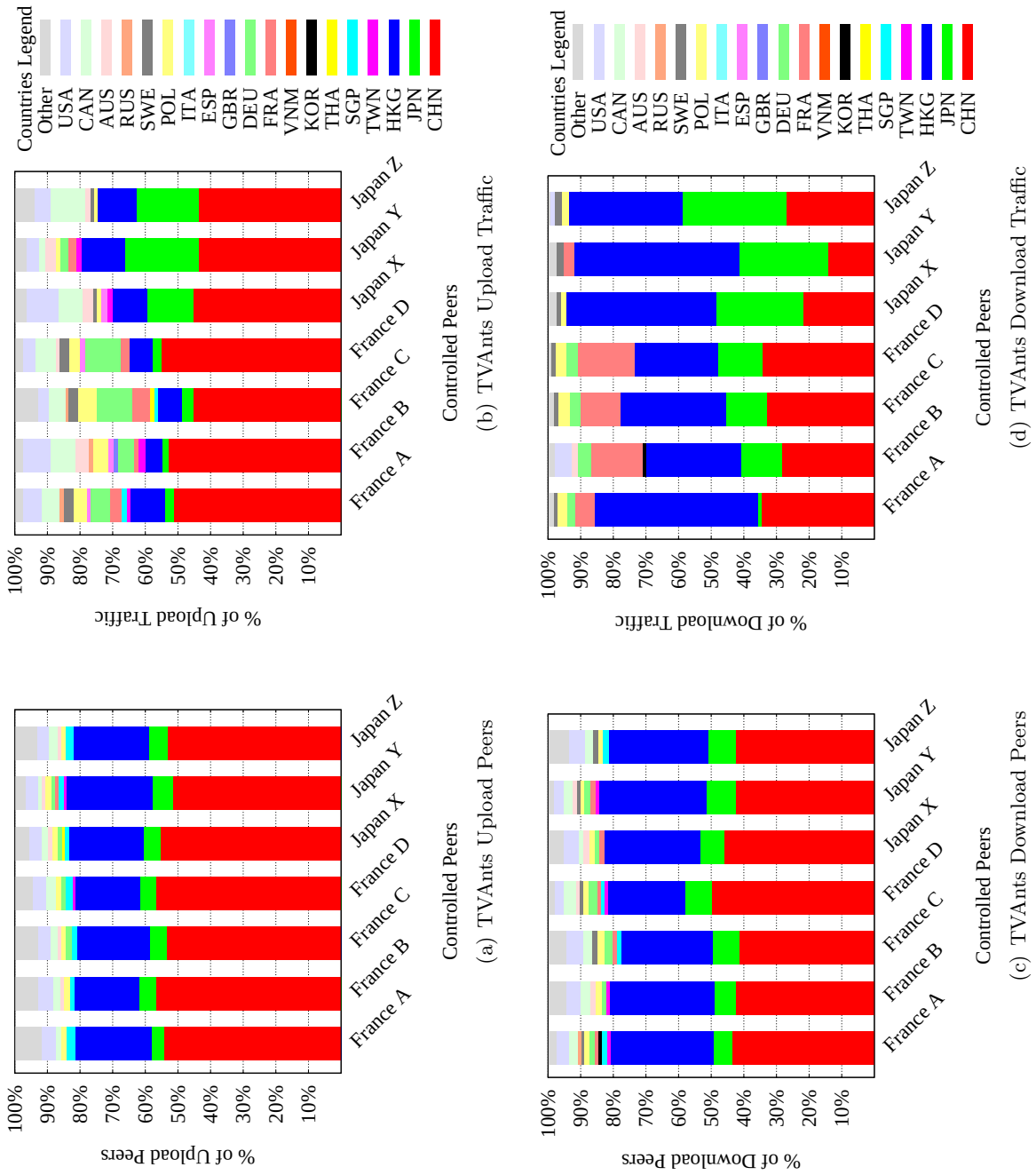
(a) TVUPlayer Upload Peers (b) TVUPlayer Upload Traffic (c) TVUPlayer Download Peers (d) TVUPlayer Download Traffic

Figure 4.11: TVUPlayer: geographic location of peers and traffic



(a) SOPCast Upload Peers (b) SOPCast Upload Traffic (c) SOPCast Download Peers (d) SOPCast Download Traffic

Figure 4.12: SOPCast: geographic location of peers and traffic



(a) TVAnts Upload Peers (b) TVAnts Upload Traffic (c) TVAnts Download Peers (d) TVAnts Download Traffic

Figure 4.13: TVAnts: geographic location of peers and traffic

4.7.4 Stability of the Traffic

In P2P systems, the end-hosts are responsible to forward the content to other. The end-hosts are not entities dedicated to stay in the network all time: they can leave or join anytime and are also prone to failures. This phenomenon is commonly referred as the churn of peers in P2P systems (chapter 2). Thus, depending on the inherent churn of peers, the P2P systems are highly dynamics and evolve continuously with time.

The information provided previously on figures 4.10- 4.13 are for the total duration of the traces. In order to point out the dynamics of the measured systems with time, we divided the traces into 5 minutes long intervals. Then, we show the evolution of the traffic by countries for each time intervals. Since all the traces have the same duration (2h45), they all have 33 time intervals. We chose these 5 minutes intervals because it was still possible to observe the dynamics of the traffic. At smaller intervals, the traffic fluctuates too largely and longer intervals fail to capture the dynamics of the systems. In order to compare with other time interval, remind that the previous stacked histograms represent a single interval of 2h45mn.

Figures 4.14 and 4.15 show the evolution of the traffic with time. For each application we present only one example in upload and download, not necessarily from the same peer from France or Japan. Even though our controlled peers from different part of the testbed were not exchanging traffic with the same countries, we just aim to illustrate the dynamics of the systems and how the traffic towards different countries fluctuates largely.

Among all the applications, the traffic of PPSTream fluctuates less compared with the other applications. However, in upload, the traffic towards France can represent 30% of the traffic (10th minute) and decreases to only 5% (70th minute) (figure 4.14(a)). The traffic of PPSTream is more fluctuating in download (figure 4.14(b)): the traffic from China counts only for 10% until 100th minute and then increases immediately to 40% at 120th minute. If we remind that SOPCast downloads the traffic from only a very few number of countries, it is interesting to observe that the amount of traffic by countries is not stable with time. For instance in figure 4.15(b), the traffic from China could range from 25% (20th minute) to 75% (75th minute) of the download traffic. The upload traffic of SOPCast fluctuates largely: a large amount of traffic to China until 10th minute, then the traffic to UK counts for a large part of the traffic until 110th minute, and finally the traffic from China becomes again the largest part of the traffic. For TVUPlayer (figures 4.14(c) and 4.14(d)), in upload, the traffic with France fluctuates largely with the time, and in download this is the case for Japan , Korea or Hong Kong. The traffic from China is still very important for TVAnts but increase largely from min 60 to min 90 in upload (figure 4.15(c)), or decreases in download from min 50 to min 80 (figure 4.15(d)).

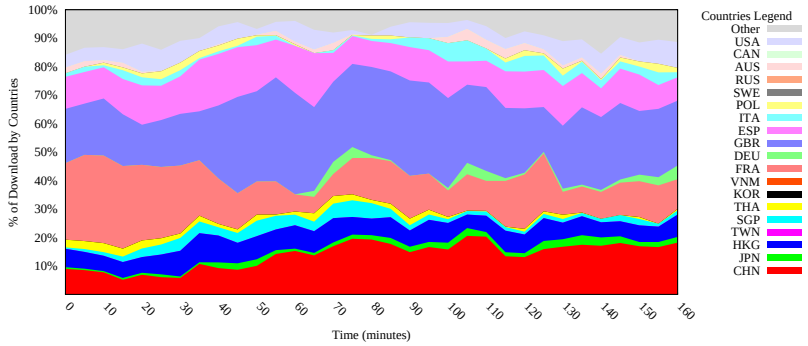
Overall, whatever the applications, the traffic is highly dynamics and fluctuates from countries to other. It illustrates the important dynamics of the P2P-TV systems. Again, our observations at the country-level could be extended to the IP level: if our nodes download a large amount of traffic from China, and then switch immediately to Europe, it is obvious that the peers providing data have also changed.

In P2P networks, the dynamics of peers will directly affect the other peers still staying in the network. If provider peers leave the network, then the remaining peers have to find other peers to get the content. For the distribution of a file over a P2P network, the departure of a peer will interrupt the download of the file for a period of time or just reduce the download speed. At the end, it will take more time for the interrupted peers to collect all the chunks of the file, and the users will just experiment a longer time to complete its download. For video live streaming, it is more crucial since the video streams impose temporal constraints. The departure of peer could lead to the loss of chunks, or the late arrival of chunks after their playback deadline. The user will immediately suffer from this situation and it will experiment a bad quality video. Since we observed previously that peers are synchronized with each other according to their position in the play-out buffer, it is more challenging for P2P-TV systems to recover from the highly dynamics of peers.

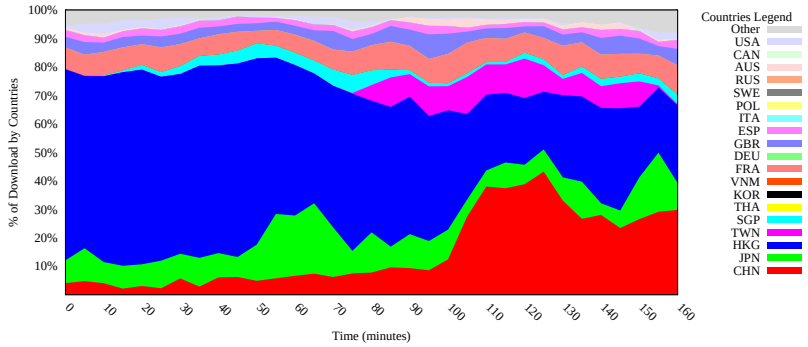
4.7.5 Discussion

From this section and the study of the geographic location of peers, we have the following observations. As it was expected, the broadcasted contents have an effect on the population of peers and their geographic location. Moreover, the current spread of these applications in Asia explains the large presence of Asian peers in our measurement. Regarding the number of peers and the amount of traffic they generate, there is not necessarily a relationship between the population of peers and the volume of traffic. We can also distinguish two distinct traits in the traffic exchanged towards different countries depending on the location of our controlled nodes. From the traffic exchanged on both directions, we find out that our nodes did not clearly reciprocate when downloading the traffic. In fact, the video streams seems to follow a directed path among peers and it is not possible for peers to reciprocate with data to their provider peers. Finally, by focusing on the traffic at smaller time intervals, we observed that the traffic is highly dynamics and fluctuates largely from countries to other. As it is a major concern for P2P systems, it is more crucial for video live streaming since the video streams impose temporal constraints.

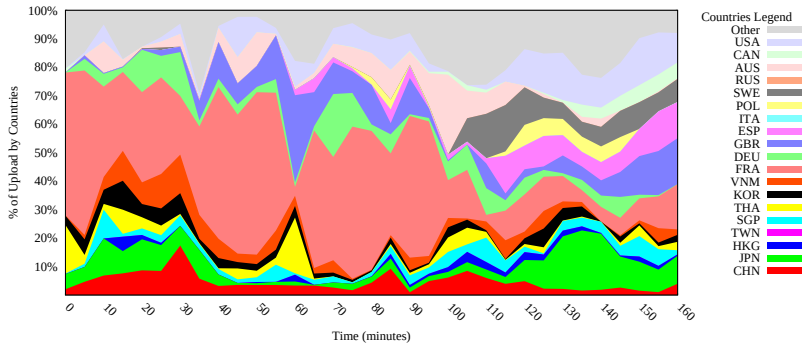
Indeed, if we remind that P2P-TV take their inspiration from the P2P file sharing system BitTorrent, multimedia flows impose temporal constraints which are not present for mere file distribution and result from the continuous nature of the transmission. Hence peers synchronize with the data each needs. Peers cannot transmit data in return, not necessarily because they are uncooperative, but because the nature of the content and the temporal constraints make it pointless. Thus, as we stated in section 4.5, fairness is not achieved in the P2P system. Since the success of BitTorrent relies mostly on the use of a mechanism to align incentives between peers in the system and allow fairness between peers, the nature of the multimedia flows obsoletes the incentive mechanism of BitTorrent. It is therefore urgent to design a new incentive mechanism adapted to the continuous nature of the multimedia flows. Such a mechanism must allow all the peers truly to collaborate in the network, even if the continuous nature of the content being distributed mitigates against the transmission of data in



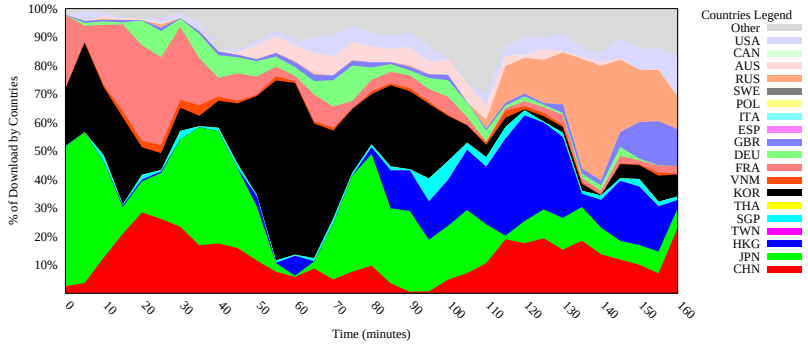
(a) PPSTream Upload



(b) PPSTream Download

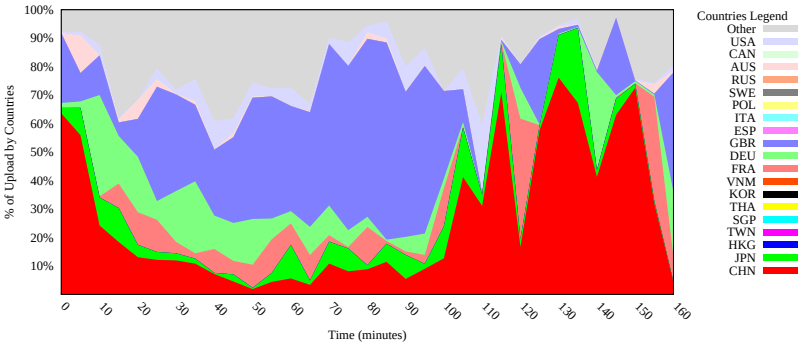


(c) TVUPlayer Upload

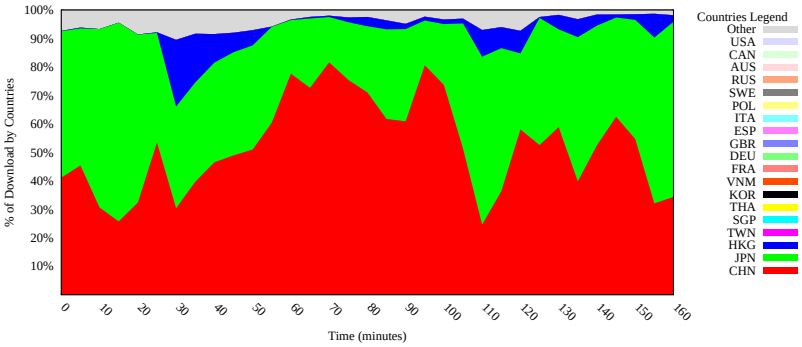


(d) TVUPlayer Download

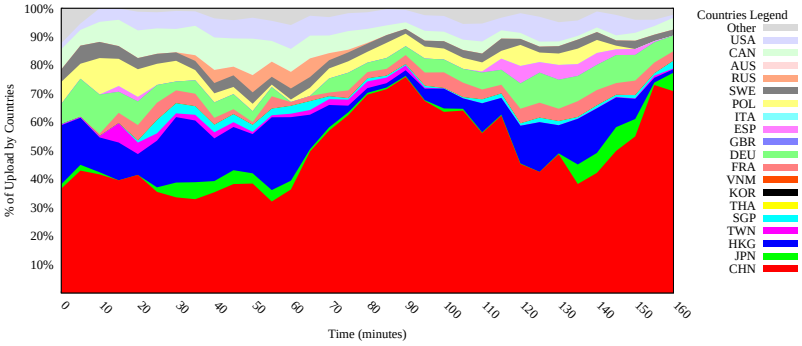
Figure 4.14: Dynamics of the traffic for PPStream (a,b) and TVUPlayer (c,d)



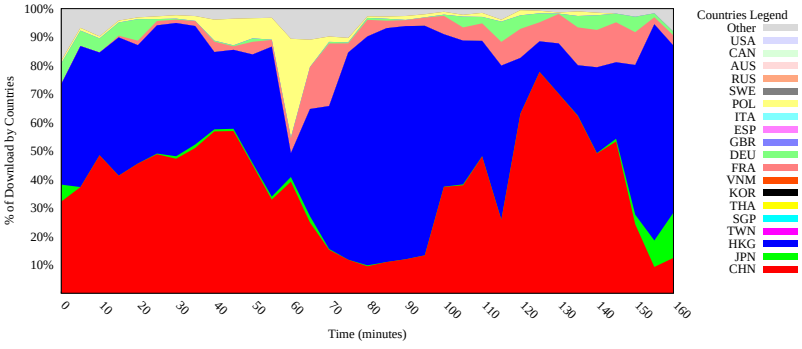
(a) SOPCast Upload



(b) SOPCast Download

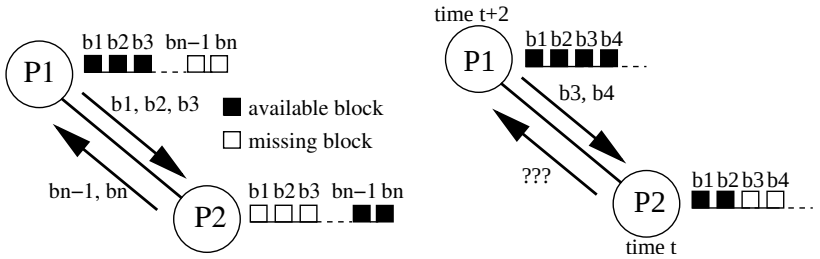


(c) TVAnts Upload



(d) TVAnts Download

Figure 4.15: Dynamics of the traffic for SOPCast (a,b) and TVAnts (c,d)



(a) P2P file-sharing (b) P2P live streaming

Figure 4.16: Chunk distribution for BitTorrent-like P2P systems

reciprocating manner. In the next section, we will present our new incentive mechanism that we designed to be best suited to P2P-TV systems.

4.8 From Measurements to a New Incentive Mechanism

From our measurement experiment, one of our main observations was the non-reciprocal exchange of data between peers that leads to unfairness in the P2P system (section 4.5). It is a crucial point regarding the scalability of P2P-TV since P2P systems rely on the fair collaboration between peers.

As we explained throughout this thesis, the P2P-TV systems adopt the mesh-based architecture inspired by BitTorrent. BitTorrent has proved its efficiency to distribute files at large scale and this is also due to its incentive mechanism used to enforce peers to collaborate in the network. As stated in chapter 2, the incentive mechanism in BitTorrent rests on reciprocal exchanges of data between peers (*tit for tat*).

Since all the blocks of a file must be downloaded before the file can be used, each block has equal importance, and peers need to recover blocks of a file in no particular order without regard for their position in the file (beginning, middle or end of file). This scenario is illustrated in the figure 4.16(a): Peer P1 has the first three blocks of the file (b1, b2 and b3), while P2 has the last two blocks (bn-1 and bn). In this situation, it is in each's interest to exchange the blocks to augment their collection of blocks.

For P2P-TV, we observed from our measurement that the video streams follow a directed path among peers. This is the reason why it is not possible to reciprocate with data to the provider peers (section 4.7.3). If we revisit the previous example (Fig. 4.16(a)), but with a multimedia flow, peer P2 has blocks bn-1 and bn and must already have received blocks before to be able to consume these ones on the fly. In contrast, P1 which has blocks b1 to b4, is not necessarily interested in blocks far ahead (in time). Its priority is to fetch blocks which it can consume soon such as blocks b5, b6 and b7, which are more important to P1 than blocks bn-1 and bn far ahead. Finally, each peer finds itself at a different point in the playout, and peers which have received blocks at playout time t+n have received more blocks and are further ahead than peers which are still at point t. Peers at playout point t+n hold blocks that are indeed of

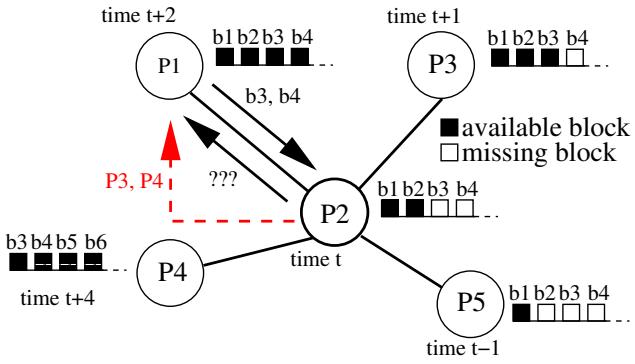


Figure 4.17: New incentive mechanism for P2P live multimedia streaming

interest to peers behind at time t. The opposite is not the case: blocks held by peers behind, at time t, are of no interest to peers at point t+n, since they have already recovered those and are moving ahead. This example is illustrated in figure 4.16(b). Peer P1 is at playout point t+2 and has blocks b1 to b4, which some are of interest to P2 (b3 and b4). P2 is behind, (at playout point t) and has nothing it can possibly send to P1.

To put this another way, P2 does not transmit data to P1, not because it is uncooperative, but because it is in the nature of the content and the temporal constraints that make it pointless. Multimedia flows impose temporal constraints which are not present for mere file distribution and result from the continuous nature of the transmission. Peers synchronize with the data each needs. If peers send data without reciprocation, fairness is not achieved. This is indeed what we observe in the course of our measurements of P2P-TV systems.

4.8.1 Towards a Novel Incentive Mechanism for P2P multimedia live streaming

For P2P-TV systems, collaboration of a peer in the P2P network cannot be evaluated merely on the basis of the quantity of data it transmits in its turn at furnishing content. This is due to the continuous nature of the media and its temporal constraints. We propose a new incentive mechanism to P2P live multimedia streaming systems which introduces a new criterion for the evaluation of the collaboration of a peer. This criterion will allow us to show whether a consuming peer invests in the exchange or not.

In our mechanism, consumer peers advertise new peers capable to offer data to provider peers. This allows providers peers to discover new sources of data. Thus, if a peer transmit no data itself to one of its provider peers, it proves it is collaborating and it is indirectly allowed to receive data. This new type of collaboration encourages peers constantly to find new peers and propose them to its data providers.

The mechanisms for finding new peers in P2P systems have already been well stud-

ied [57] and work as follows: each peer can add information which it sends to its neighbors (with its *buffer map*) a list of peers it knows about. Peers forward randomly the list of peers that they have discovered, in a analogous manner to the protocols used for epidemic diffusion (*gossip protocols*).

Figure 4.17 illustrates the new incentive mechanism we propose. Peers P1 and P2 are in the same configuration as before (Fig. 4.16(b)). Remember that BitTorrent creates a mesh between the peers which own various blocks of data. In our example, P2 knows P1, P3 and P4. They are ahead the playout time of P2 so they have blocks which are interesting for P2 (P1 is at time t+2, P3 at time t+1 and P4 at time t+4). P2 provides also P5, which is behind (at time t-1). P1, P3, P4 and P5 constitute the neighborhood of P2. They all have their own neighborhood, but for clarity reasons, it is not shown in the Figure.

Since P1, P3 and P4 are situated ahead the playout point of P2, it is highly likely that P1, P3 and P4 are situated at close playout points and are looking for common blocks of data. Because P1, P3 and P4 have their own neighborhood with different content provider peers, these three peers also hold blocks possibly interesting to others. This is the case for P4 that has interesting blocks for P1 (b5, b6). Thus, peers P1, P3 and P4 are certainly interested to establish peering relationship together to exchange blocks of data, but they still do not know each other.

Even though the continuous nature of the media means that P2 is not able to offer data to P1 in return (solid black arrows in Figure 4.17), P2 can collaborate in the system by transmitting to P1, information about the new potential sources of data (red-dashed arrow in Figure 4.17). P2 is incentivized to find these new pairs and report them to providers.

In our example, P2 informs either P3, P4 or even P5 (or both) about P1. The reaction of P1 according to the new discovered peers is as follows:

- P4 holds blocks which interest P1. P1 and P4 have to organize themselves to exchange blocks of data. P1 has to convince P4 to send it data, by finding new blocks or new sources of data. However, P2 proposed a new source of data to P1, proving its fair collaboration. P1 must grant P2 in return by continuing to send data to P2. P1 may also increase the amount of data it sends to P2 to show its gratitude.

- P3 does not hold blocks which interest P1 but P1 can deduce, with the *buffer map*, that P3 is a provider to P2. Even though the information does not let P1 discover a new source of data, P2 collaborates actively and deserves to continue to receive data from P1. P1 can also limit the delivery of data towards P2, notably if the resources of P1 are used by many peers.

- If the announced peers (P3 or P4) are already known by P1, P1 will have the same reaction as the previous case. P2 collaborates actively and P1 continues to send it data. However, the information was not relevant for P1, which can limit the amount of data it sends to P2.

- If P2 informs P1 about P5; P5 has no block of interest for P1, and is in addition, behind P2, P1 estimates P2 has not made enough effort to find new peers. P1 can then sanction P2 by decreasing the amount of data it sends to P2 or by canceling their peering relationship.

To summarize our proposal, in order to enable peers to truly collaborate in P2P-TV network, we proposed a new incentive mechanism in which the evaluation of peers' collaboration is based on the control information exchanged rather than only data. Indeed, this sole criterion is not appropriate to systems for distribution of live multimedia flows because of the nature of continuous media, and their temporal constraints. This mechanism matches our observations of real P2P-TV systems deployed on the Internet and allows to all peers that cannot reciprocate in the exchange of data to collaborate in the P2P network.

In order to have a fair validation of the proposed mechanism, we will implement it in a commonly used simulator such as PeerSim [128]. We will focus notably on the overheads associated with the control information. We must refine the parameters of the mechanism with regard to the reactions of data providers and the quality of peers reported. The mechanism must equally take into account the reputation and confidence in the relationship between peers.

4.9 Summary

Since our previous measurement experiment, there have been many other measurement studies of P2P-TV. At the beginning of this chapter, We surveyed most of them and use a classification related to the initial knowledge of the studied systems. As for our measurement experiments, some related works passively measured the traffic generated by P2P systems and infer their main mechanisms and properties. Among all these studies, the works of Hei [84] or Ali [93] are very close to ours. However, they only focus on a single application or did not measure events that exhibit interest to be watched in live.

In this chapter, to extend the scope of our previous experiment, we performed a new measurement experiment of P2P-TV at larger scale. It has been conducted in order to obtain a more global view of the measured P2P network. To this end, we measured the P2P-TV traffic with multiple measurements points situated in distinct locations in France and in Japan. Again, we measured the traffic of popular P2P-TV applications watching live soccer games.

We wanted to characterize more precisely the P2P-TV overlay network and the main properties of these systems. Among interesting topics, we focused on the organization of peers, the amount of traffic they exchange together and their level of collaboration. It is an important concern regarding the scalability of these P2P systems. By collecting the traffic at distinct places, we were also interested to study whether there are differences between the traffic collected in France or Japan. As it is crucial for the point of view of ISPs, we also investigated the locality of the traffic, the distance between peers and their geographic location.

As a first step to characterize the organization of peers, we observed that our French nodes get the content at a further distance than the Japanese nodes. It is an important issue since long distance may occasion large end-to-end delay, leading to a bad video streaming quality. Thus, in order to study the locality of the traffic, we showed that many countries or ASes are involved in the traffic exchanges, and this is especially the case for the upload traffic in which we observed hundreds ASes (more than 200 for PPSTream). By inferring the distance in number of AS hops to exchange the traffic (i.e. the number of ASes in the path from one of our node to a remote peer), we pointed that the downloaded traffic gets across a larger number of ASes than the upload traffic. In our opinion, the traffic is not kept local and is exchanged arbitrary towards many distinct locations. This is an important issue because it can consume the network resources by overloading the links between ISPs, leading to probable additional costs to transport the traffic.

Regarding the countries involved in the traffic, as it was expected, the broadcasted content has a strong impact on the geographic location of peers present in the traces. For instance, a larger number of Japanese peers were present when the games involved Japan. However, this is not a rule and we observed many peers from UK during a game between China and Japan. Moreover, some countries can count a large number of peers but they did not contribute in the traffic whereas the opposite can be the case (a few number of peers for a large amount of traffic). As an example, there is a few number of Korean peers in TVUPlayer (5% for TVU_Jz) but they count for a large amount of the upload traffic (30%). Differently, a large number of peers such as Chinese peers in SOPCast (40% for SOP_Fb) did not necessarily leads to a large amount of traffic (8%). Overall in the traces, only 25% or 30% of peers are really exchanging traffic and contribute to 90% of the total traffic.

We also observed different behaviors according to the applications or the different locations where the traces have been collected. Some applications have roughly the same population for the French and Japanese traces (PPSTream) while other applications present two different population of peers in the France or Japanese traces (SOPCast).

Moreover, it seems that for all the applications, the population in download is different from those in upload. This is especially the case in the traces collected in France where the download traffic involved many Asian countries while the upload traffic is towards European countries. In fact, the traffic seems to be spread into the overlay following a directed path. This could explain the level of contribution between peers. Indeed, we observed that the peers did not exchange traffic in a reciprocal manner and fairness is not achieved in these P2P-TV systems. The fact is that peers are not necessarily uncooperative but the continuous nature of the content and the temporal constraints prevent them to reciprocate with data. Peers are synchronized with the data each needs and cannot transmit data in return.

Remind that P2P systems rely on the collaboration between peers that share their resources to forward the data to other. In the case of P2P-TV, the nature of the video streams prevents them to retransmit data to their provider peers. Therefore, we proposed a new incentive mechanism adapted to multimedia live streaming to enforce

peers truly to collaborate in the network even if the continuous nature of the content being distributed mitigates against the transmission of data in reciprocating manner. This mechanism matches our observation of real P2P-TV systems deployed on the Internet.

After all, this measurement study at larger scale allows us to better characterize the P2P overlay network. Throughout all this study, we quantify relevant parameter for these systems: at the network level, we provide information on the overall volume of traffic, number of packets and ratio of transport protocol. At the peer level, we indicate the number of peers in each trace and discuss the possible number of peers in the network. We also show the amount of data each peer send or receive, and the ratio of traffic the peers share in the network. For network operators (ISPs), we provide useful results regarding the locality of the traffic, the number of ASes or countries involved in the exchange. Furthermore, at the user point of view, we study the geographic location of peers, and the usage of these applications. All these parameters are of interest for future simulations, emulation or models of these systems.

Chapter 5

Conclusion and Perspectives

From the World Wide Web service and hypertext documents in the early 1990s to current popular services such as file-sharing, VoIP, online gaming or video services including television and video-on-demand, the Internet has become the main medium to deliver all kinds of content to users. In particular, these last years, there have been tremendous efforts to design new architectures to enable the Internet to deliver video streams at large-scale and, thus, to offer a television service to the Internet users. These new architectures rely on the use of the P2P paradigm, which showed its efficiency to distribute content at large-scale.

We surveyed the main P2P-TV architectures that we classified into three categories depending on their overlay construction: tree-based including single-tree and multiple-tree, or mesh-based. Even though the mesh-based architecture has been commonly admitted as the architecture of reference, these architecture proposals have never been really implemented, experimented nor deployed on the Internet. In fact, most of these proposals have been evaluated by performing trackable simulations of the systems. It is therefore essential to provide realistic input parameters in order to obtain accurate simulation results. In parallel, many commercial P2P-TV applications have been released on the Internet. These applications are proprietary and do not provide any architecture or implementation details.

Overall, at the beginning of this thesis, the P2P-TV systems were largely unknown, and there were no information available on these systems. Even simple but essential parameters that are needed to simulate P2P-TV systems were still undetermined such as the lifetime of peers in the network, the amount of data they send and receive or the neighborhood of a peer.

We have therefore followed a new approach to conduct our research on P2P-TV. In this thesis, in order to fill the gap between the academic proposals and commercial architectures, we performed extensive measurement of popular P2P-TV systems. By measuring the network traffic of P2P-TV applications, we aimed to get a better understanding of these systems, infer their main mechanisms and properties and study the

impact of this traffic on the networks. Thus, we analyzed the network traffic generated by four of the most popular P2P-TV applications collected in our campus network.

The results can be useful to identify the traffic generated by such applications, to understand the impact of the traffic on the network and to build realistic simulations of these systems.

We outlined similarities and differences among P2P-TV applications in terms of the transport layer protocols and the related ports they use. We gained some knowledge regarding statistical properties of this traffic (e.g. Packet Size, Inter Packet Time distribution). Furthermore, this study allows understanding how traffic from peers participating to a P2P-TV network is divided into upload and download, signaling and data. Looking at packet size statistics, packet-rates, and scaling properties of this traffic allows building better simulations and understanding better the impact of these applications on the network.

Looking at the peers interacting with our controlled nodes, we also inferred some knowledge regarding peer behavior in P2P-TV. We derived information on the network of peers distributing real-time content, noticing that the number of peers with which our nodes were able to exchange significant amount of data was quite low. We also studied the behavior of the top video downloaders from our nodes and the evolution of peers' neighborhood in time for all the applications. We also inferred different download policies used by the applications, revealing a different design of the considered architectures. Finally, we analyzed the durations of the peer connections. This information is useful as a reference model for simulations.

We believe that, while the results of our study can be useful for simulating new architectures using more realistic input parameters, such applications are rapidly expanding and they should be constantly monitored.

Following our first measurement experiment, there have been many other measurement studies of P2P-TV. We presented most of these related studies and classified them according to their initial knowledge of the measured system. As for our previous experiment, most of these measurement studies have a local view of the P2P-TV and they focus only on their mechanisms and their architectures. In order to extend the scope of our previous experiment, we performed a second measurement at a larger scale by multiplying the measurement points which were situated in France and in Japan. Thanks to this experiment, we got a more global view of the P2P network and it allows to better characterize the overlay network and the organization of peers.

We showed that the distance between peers are longer for peers situated in France than those in Japan and it can lead to a bad video quality for these peers. Then, we discovered that the traffic was exchanged towards hundreds of ASes and countries. It makes us assume strongly that there is no mechanism to keep the traffic local. It is a challenging issue for ISPs to perform traffic engineering tasks in order to optimize their routing and their cost to provision their network. We studied the geographic location of peers and the volume of traffic by countries and showed that the population of peers depends obviously on the broadcasted contents. We point out that our peers encountered a large number of peers during their video sessions, but only a subset of these peers really contribute to a significant part of the traffic. Surprisingly, some

application as PPSTream has the same geographic distribution of peers in all the traces while other applications as SOPCast exhibits distinct distribution for our Japanese and French peers. Among all the encountered peers, only 25% or 30% are really exchanging traffic and contribute to 90% of the overall traffic. Moreover, we observed that peers are mainly producers or consumers, but they do not exchange data reciprocally with each other. The fact is that the data are spread into the overlay following a directed path. This is due to the continuous nature of the video streams that makes peer are synchronized with each other and cannot reciprocate with data. Thus, fairness is not achieved in the system and it is an important concern for the scalability of these P2P systems.

Remind that the starting point of all our study was to get knowledge of P2P-TV, we believe our measurement experiments reach their goals for the understanding of these systems. As a major perspective for this thesis, we will introduce our new incentive scheme that matches our observations of P2P-TV systems and it is designed to enforce peers to truly collaborate in the network even if the continuous nature of the video streams make it pointless. Besides, many issues are still being open (e.g. copyright of exchanged content [129, 130], performance of P2P-TV over wireless networks [131, 132], diffusion of broadband connections [133, 134], quality of experience [135, 136], standardization [137]).

Finally, during all this thesis, we obtained valuable information on P2P-TV by performing passive traffic measurement. We started our measurements in our campus network with a local view of the systems and infered their main mechanisms and properties. Then, we performed an experiment at larger scale by multiplying the measurement points in order to obtain a more global view of the P2P network and to better characterize the overlay.

As a perspective of our works, we are currently collaborating with Zattoo, a company that released its own proprietary P2P-TV system. This research is currently at its early stage and seems to be very promising. By monitoring continuously their system, we know in real time the number of users in the systems or by channels, the amount of data exchanged between peers and numerous statistics about their system. Thanks to our collaboration with Zattoo, we aim to investigate a system with a global knowledge. We are expecting to confirm or enhance our previous observations of P2P-TV and extend them in order to improve their system.

Based on our measurement, we propose a new incentive mechanism that matches our observations of real P2P-TV systems. Evaluating this kind of mechanism is very challenging. As a future work, we aim to implement this mechanism in a commonly used simulator (such as PeerSim) to have a fair validation of the proposed mechanism. Thanks to our knowledge of P2P-TV systems, we will be able to simulate realistic environment and scenarios for such systems.

Appendix A

Thesis Publications

International Journals

Traffic Analysis of P2P IPTV Communities
T. Silverston, O. Fourmaux, A. Dainotti, A. Botta, A. Pescape, G. Ventre and K. Salamatian
in Elsevier Computer Networks, March 2009

International Conferences

Towards an Incentive Mechanism for P2P Multimedia Live Streaming Systems
T. Silverston, O. Fourmaux and J. Crowcroft
in the proceedings of IEEE International Conference on Peer-to-Peer Computing (IEEE P2P'08), short paper session, September 2008

Measuring P2P IPTV Traffic on Both Sides of The World
T. Silverston, O. Fourmaux, K. Salamatian and K. Cho
in the proceedings of the student workshop of the 3rd International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT'07), December 2007

Measuring P2P IPTV Systems
T. Silverston and O. Fourmaux
in the proceedings of ACM 17th International workshop on Network and Operating Systems Support for Digital Audio & Video (ACM NOSSDAV'07), June 2007

P2P IPTV Measurement: A Case Study of TVAnts
T. Silverston and O. Fourmaux
in the proceedings of the student workshop of the 2nd International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT'06), December 2006

National Conferences

Mesures d'applications P2P-TV de part et d'autre du monde
T. Silverston, O. Fourmaux, K. Salamatian et K. Cho
au Colloque francophone sur l'ingénierie des protocoles (CFIP'09), Octobre 2009

Caractérisation du trafic P2P IPTV : une analyse d'échelle
T. Silverston, O. Fourmaux, et K. Salamatian
au Colloque francophone sur l'ingénierie des protocoles (CFIP'08), Mars 2008

Annexe B

Version française

B.1 Introduction

Ces dernières années, les utilisateurs de l'Internet ont eu un impact important sur l'évolution et les services de ce réseau. Les utilisateurs créent eux-mêmes de nouveaux contenus et sont à l'origine de nouveaux services. Ils s'organisent en communautés pour lesquelles une participation active, l'interaction entre utilisateurs et le partage d'information sont grandement encouragés et requièrent de nouvelles technologies pour les soutenir. Les nouvelles demandes des utilisateurs ainsi que les nouvelles formes d'interaction dirigent l'évolution du réseau, amenant de nouvelles applications, de nouveaux paradigmes de communications et de nouvelles architectures réseaux. Pour illustrer ce profond changement citons notamment l'explosion des blogs Internet, les systèmes de distribution et publication de vidéos, les réseaux sociaux construits sur le web, les mondes virtuels, les jeux en réseaux et pleins d'autres encore [1, 2].

En interagissant à travers le réseau, les utilisateurs créent de nouvelles formes de communautés et de nouvelles formes de distribution de contenu. Nous assistons à un changement du paradigme de distribution traditionnel avec peu de fournisseurs et beaucoup de consommateurs vers un nouveau paradigme comportant beaucoup de fournisseurs de contenu et tout autant de consommateurs [3].

De plus, la disponibilité de nouveaux services et formes d'interaction impulsée par les utilisateurs modifient, dans le même temps, les comportements des utilisateurs et leurs attentes. Les gens commencent à utiliser l'Internet pour des activités qui avaient lieu précédemment dans d'autres contextes et à l'aide d'autres technologies. C'est le cas, par exemple, de la télévision par Internet (IPTV) et les jeux en réseaux dans les mondes virtuels. Les moments et les lieux pour se livrer à ces nouvelles activités ont également changé et ces nouveaux services deviennent omniprésents. Les personnes ne souhaitent plus uniquement utiliser les services fournis par Internet à leur domicile mais souhaitent en bénéficier dans tous les lieux qu'ils fréquentent comme leur lieu de travail ou d'autres lieux publics comme des cafés ou restaurants. De plus, les utilisateurs interagissent

entre eux et s'organisent en grandes communautés de taille planétaire sans rapport avec leur localisation géographique. De tels nouveaux comportements rendent obsolètes les systèmes de distribution de contenu traditionnels, ce qui encourage les opérateurs réseaux et l'industrie en général à concevoir de nouvelles architectures pour supporter ces services. Tous les phénomènes mentionnés participent à la grande convergence de tous les réseaux de distribution de contenus actuels (Télévision, Téléphonie, et Donnés) en un unique fédérateur : l'Internet.

La diffusion de la télévision sur l'Internet est amenée à se développer lors des prochaines décennies et à devenir de plus en plus populaires pour plusieurs raisons [4]. Tout d'abord, il est maintenant admis que pour des évènements comme les attentats du 11 Septembre 2001, l'Internet a été l'une des principales sources d'informations pour les personnes qui se trouvaient sur leur lieu de travail. De plus, les utilisateurs semblent délaisser les chaînes de télévision généralistes pour préférer regarder les chaînes de TV spécialisées ou celles qui les permettent d'interagir avec d'autres utilisateurs ou d'ajouter eux-mêmes du contenu (commenter ou poser des questions à la communauté de personnes regardant la même vidéo est un exemple typique) [5]. Une autre raison est que pour certains pays, l'offre télévisuelle et sa qualité sont souvent insuffisantes. Enfin, le phénomène de diffusion de chaîne de télévision personnelle est en constante augmentation ; cela est dû à l'arrivée des plateformes de diffusion vidéos de type "**Tube*" telles que Youtube permettant la création de TV personnelles toujours plus élaborée.s En effet, de nombreuses sources rapportent de la perte d'audience et de revenue des chaînes de télévision traditionnelles et des industries publicitaires. Ainsi, il est très important de comprendre ces nouvelles technologies afin de les supporter et de les améliorer [9].

L'utilisation du concept de pair à pair (P2P) pour diffuser de la télévision en direct sur l'Internet (P2P-TV) est devenue une alternative légitime aux autres approches que sont le modèle classique client-serveur, les réseaux de distribution de contenu [11] ou les communications multipoints de niveau réseau (IP-multicast). De plus, en utilisant l'infrastructure de l'Internet existante comme un médium de diffusion et en exploitant la participation des utilisateurs pour la création de réseaux de distribution de contenu, la technologie P2P pour la diffusion de télévision en direct sur l'Internet a les potentiels pour : (i) rendre n'importe quelle chaîne de télévision d'un pays disponible pour le monde entier, (ii) transformer chaque utilisateur d'Internet en un créateur de contenu, capable de diffuser sa chaîne de télévision personnelle à des coûts négligeables.

Les raisons évoquées expliquent la popularité croissante de ces applications auprès des utilisateurs de l'Internet. Toutes ces applications émergentes amènent de nouveaux challenges que nous allons présenter dans la section suivante.

B.1.1 Challenges

Dans ce manuscrit, nous nous intéressons aux systèmes pairs-à-pairs de diffusion de télévision en direct sur l'Internet (P2P-TV). Plus exactement, nous avons mesuré ces systèmes et analysé leur trafic réseau afin d'en comprendre le fonctionnement et d'en déduire les principaux mécanismes.

Au début de nos travaux de thèse, nous avions remarqué que beaucoup d'architec-

tures P2P avaient déjà été proposées pour diffuser des flots vidéos en direct. Pourtant, seules quelques-unes d'entre elles ont réellement été implémentées ou expérimentées [12, 13] mais à petite échelle par rapport à l'Internet. La plupart de ces propositions utilisent des simulations pour valider l'architecture proposée et ses mécanismes. Dans le même temps, de nombreuses applications commerciales ont été déployées sur l'Internet [14–20] sans dévoiler aucun détail sur leur architecture et leur implémentation. Ainsi, il y avait une méconnaissance totale de ses systèmes P2P-TV et aucune information fiable n'était disponible pour pouvoir les évaluer correctement. Dans ces conditions, la simulation rigoureuse de ces systèmes était des plus délicate en raison du manque d'information indispensable : paramètres réalistes de ces sytèmes, scénarios, ou même d'outils de simulations adaptés à la nature continue des flots diffusés. Par ailleurs, une connaissance précise des systèmes P2P-TV aiderait à mettre en évidence leurs défauts pour pouvoir ensuite les étudier et les corriger.

Ainsi, nous avons adopté une approche originale pour conduire nos travaux de recherche. Au lieu de concevoir de nouvelles architectures ou mécanismes sans se soucier du fait qu'ils soient réalisables sur l'Internet, nous nous concentrons d'abord sur les applications commerciales qui ont été déjà déployées avec succès sur l'Internet. Notre premier objectif est d'obtenir une meilleure compréhension des systèmes P2P-TV. Nos observations et autres découvertes devraient également nous permettre d'évaluer ces systèmes, de fournir des informations pour de futures modèles ou simulations réalistes et proposer de nouveaux paradigmes ou bien mécanismes pour améliorer ces systèmes. Notre approche se base sur les observations "ìn vivo" de systèmes réels sur l'Internet, et non pas sur des suppositions "in vitro" de l'Internet pour proposer de nouvelles architectures.

Puisque les applications commerciales sont propriétaires et que leur code source n'est pas disponible (libre), nous pouvons seulement avoir accès à ces systèmes en les mesurant pour déduire leurs mécanismes sous-jacents. La seule façon de le faire est de collecter leur trafic réseau et de l'analyser. Pour cela, nous avons effectué d'intensives expériences de mesures d'applications P2P-TV commerciales. Cette approche externe, en considérant les applications comme des "'boîtes noires", est la première étape pour combler l'écart des propositions dites "académiques" avec les applications "'commerciales".

En outre, la popularité des applications P2P-TV est grandissante et compte déjà des millions d'utilisateurs partout dans le monde. Avec le succès des applications de partage de fichiers (BitTorrent, eDonkey [21, 22]) ou bien des services de diffusion de vidéo (Youtube, Dailymotion [23,24]), les trafics P2P et vidéo comptent déjà à eux deux pour la plus importante part du trafic total de l'Internet [25]. Il est donc attendu que le trafic P2P-TV va contribuer à augmenter encore le trafic de l'Internet et comptera pour une part significative de ce trafic. Il est également intuitif de comprendre que le comportement des utilisateurs des applications P2P-TV aura un effet sur le trafic généré. Par exemple, les utilisateurs devraient se comporter comme avec la télévision traditionnelle et changer souvent de chaînes (*channel zapping*) ce qui aura un effet sur le trafic généré.

B.1.2 Contribution de la thèse

Dans cette thèse nous avons consacré notre recherche à l'étude de systèmes P2P-TV. Plus précisément, nous avons effectué des expériences de mesure et étudié le trafic réseau généré par les applications P2P-TV les plus populaires : PPLive, PPSTream, SOPCast et TVAnts. L'étude de plusieurs applications en même temps rend notre analyse plus complète et nous permet de généraliser les résultats obtenus.

Dans un premier temps, nous cherchons à mieux comprendre les mécanismes utilisés par ces applications et leur impact sur le réseau en regardant directement le trafic qu'elles génèrent, et ce, en dépit de leur utilisation de protocoles non documentés. Ces travaux présentent l'analyse détaillée du trafic réseau généré par les applications P2P-TV, lequel trafic a été collecté dans le réseau de notre université. Dans un premier temps, nous distinguons le trafic de signalisation et celui vidéo, et examinons les propriétés statistiques de ces différents trafics. Nous avons également étudié le trafic dans les deux directions que sont le trafic montant (*upload*) et descendant (*download*) afin d'en caractériser les principales propriétés séparément. Nous déterminons quels sont les protocoles de transport utilisés et quelles sont les conséquences de ces différents choix. Nos résultats montrent que le trafic P2P-TV est assez différent du trafic d'autres applications réseaux et qu'il est difficile de l'identifier. De plus, nous révélons quelques caractéristiques essentielles de ce trafic pour en comprendre l'impact sur le réseau. Ensuite, nous nous intéressons à l'étude du comportement des pairs, c'est-à-dire comment les pairs interagissent entre eux, comment ils contribuent à la diffusion du contenu et quelles sont les politiques de téléchargement des données des différentes applications. Nous observons également le phénomène de "peer churn", c'est-à-dire l'arrivée et le départ incessants des pairs dans le réseau et étudions la durée de vie typique d'un pair dans le réseau.

Grâce à cette double analyse du trafic, nous améliorons notre connaissance des caractéristiques du trafic de ces nouvelles applications et du comportement de leurs utilisateurs. Les résultats que nous présentons sont utiles pour identifier le trafic généré par ces applications, pour en comprendre l'impact sur le réseau et construire des simulations réalistes.

À la suite de notre première expérience de mesure, nous voulions généraliser la portée de notre étude. Notre première expérience de mesure se concentrait sur les propriétés locales des systèmes P2P-TV comme la caractérisation de leur trafic ou les mécanismes qu'ils utilisent. Par la suite, nous souhaitions plutôt caractériser le réseau P2P lui-même, et l'organisation des pairs dans le système global.

Nous avons donc effectué une seconde expérience de mesure à plus large échelle afin d'obtenir une vision globale des réseaux P2P mesurés. Plutôt que de mesurer les systèmes P2P avec un unique point de mesure, nous les avons mesurés avec de multiples points de mesures situés de part et d'autre du monde, en France et au Japon. En se basant sur les mesures de cette seconde expérience, nous discuterons principalement du niveau de collaboration entre les pairs et de la localité du trafic. En effet, le trafic P2P est devenu un problème important pour les fournisseurs d'accès à Internet (FAI). L'échange réciproque des données entre pairs est aussi une condition essentielle pour

le passage à l'échelle de ces systèmes P2P. De plus, entre autres sujets d'intérêt, nous étudions l'organisation globale des pairs dans le réseau, la quantité de trafic qu'ils échangent et le niveau de collaboration des pairs. Nous nous intéresserons aussi à la distance entre les pairs, la situation géographique des utilisateurs et les effets du contenu diffusé sur les communautés d'utilisateurs d'applications P2P-TV.

B.1.3 Structure du document

Ce manuscrit de thèse est divisé en cinq chapitres qui s'articulent de la façon suivante. Dans le chapitre 2, nous présentons tout d'abord les différentes techniques pour délivrer des flots vidéo en direct sur l'Internet. La deuxième partie de chapitre décrit l'état de l'art des architectures P2P de diffuson de flots vidéo en direct. Dans le chapitre 3, nous présentons notre première expérience de mesure d'applications P2P-TV et notre analyse de trafic. Nous concentrons cette étude sur les caractéristiques du trafic généré, la compréhension des mécanismes utilisés par les applications P2P-TV et le comportement de leurs utilisateurs. Afin de généraliser la portée de nos expériences, le châpitre 4 présente notre deuxième expérience de mesure effectuée à large échelle afin de mieux caractériser les réseaux P2P dans leur globalité. Le châpitre 5 conclut le document en résumant les travaux présentés dans cette thèse et en présentant les perspectives futures.

B.2 Architectures P2P de diffusion de flots multimédia en direct : État de l'art

De la distribution de contenus tels que des objets web ou des fichiers à la diffusion de flots multimédia en continue (*multimedia streaming*), les services de distribution de contenu sur l'Internet évoluent en fonction des besoins des utilisateurs et de la technologie. Le streaming est souvent défini comme étant la lecture en continue de flots de données multimédia comme ceux audios ou vidéos [26]. Á la différence des fichiers qui nécessitent d'être entièrement téléchargé avant d'être lus, les flots de *streaming multimédia* sont lus au fur et à mesure de leur réception, ce qui permet aux utilisateurs d'apprécier le contenu immédiatement.

Par *streaming*, on peut distinguer le streaming en direct – comme la télévision – du service de vidéo à la demande (VoD). Les services de VoD et de vidéo en direct sont chacun des services de streaming mais pour la VoD, le contenu diffusé est un fichier pré-enregistré dont la diffusion est à la demande de l'utilisateur (films, émission de télévision) alors que le streaming de vidéo en direct diffuse un contenu qui présente un véritable intérêt à être regardé en direct (comme les évènements sportifs, les chaînes de télévision, les concerts) et qui n'est pas à priori stocké sur un serveur. Dans nos travaux, nous nous concentrons uniquement sur le service de streaming vidéo en direct comme c'est le cas pour la télévision.

Les applications de diffusion de vidéo en direct comme la télévision sont actuellement en plein essor sur l'Internet (Internet-TV) et l'on s'attend à ce qu'elles soient massivement utilisées dans le future. Pareil à la télévision traditionnelle (hertzienne, satellite), les applications de diffusion de télévision sur l'Internet ciblent un grand nombre d'utilisateurs en même temps. De l'architecture classique client/serveur communément utilisée sur l'Internet aux architectures pairs-à-pairs, il existe de nombreuses propositions pour permettre à l'Internet de se doter d'un service de diffusion de vidéo en direct. Parmi celles-ci, citons par exemple les réseaux de distribution de contenu ou les communications multipoints de niveau réseau (IP-multicast). Actuellement, les architectures P2P sont considérées comme étant les solutions les plus adaptées pour offrir un service de streaming vidéo en direct sur l'Internet car le modèle P2P permet d'atteindre de très larges audiences comme celles requises par la télévision.

Pour le résumé en français de cette thèse, nous présentons les autres propositions et expliquons en quoi elles ne sont pas adaptées pour offrir un service de télévision à l'Internet. Ensuite, les architectures P2P de diffusion de flots multimédia en direct sont évoquées. Une présentation détaillée de ces architectures est disponible dans la version anglaise de cette thèse.

Pour diffuser des flots multimédia en direct sur l'Internet, une solution basée sur le simple modèle client/serveur où le serveur source est le seul responsable de la duplication du contenu à travers de nombreuses communications unicast (un-vers-un) vers tous les récepteurs ne passera pas à l'échelle. En effet, les serveurs ont des ressources limitées (CPU, bande passante), lesquelles décroissent proportionnellement avec

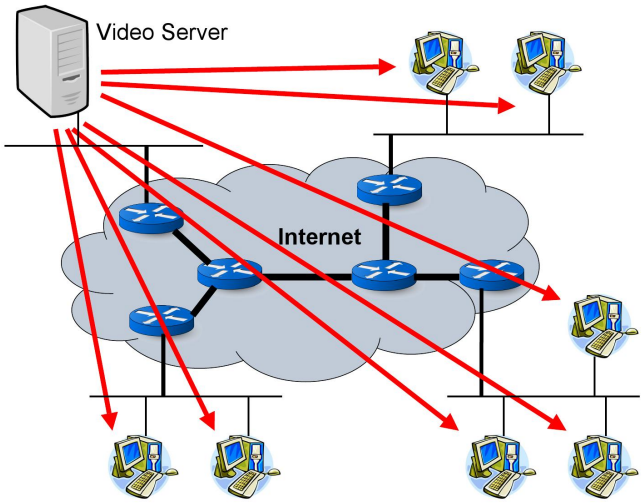


FIG. B.1 – Client-serveur

le nombre d'utilisateurs. Cela est illustré sur la figure B.1 et cette solution conduira au problème inhérent du goulot d'étranglement du côté du serveur. Quand le serveur atteint ses limites, il ne lui est plus possible de servir de nouveaux arrivants. Cette solution ne passe donc pas à l'échelle du nombre des utilisateurs.

L'utilisation de serveurs répliqués via les réseaux de distribution de contenu [11] (CDN) pourrait étendre le modèle client/serveur. Dans cette solution, comme cela est indiqué sur la figure B.2, le serveur source envoie la vidéo à des serveurs du CDN situés de façon stratégique proche des utilisateurs en bordure de réseau. Les clients ne téléchargent plus la vidéo de la source mais sont redirigés vers un serveur du CDN proche d'eux. Un des avantages des CDN est de déplacer la charge de trafic du coeur du réseau vers la bordure du réseau. L'utilisation de CDN raccourcit aussi le délai vers les utilisateurs. Cependant, si l'on souhaite atteindre un nombre important d'utilisateurs, de nombreux serveurs doivent être déployés ce qui entraîne un coût d'infrastructure très important. Plus le nombre de clients est important, plus la bande passante du serveur vidéo source et des serveurs du CDN doit augmenter proportionnellement. Par exemple, une plateforme de diffusion vidéo comme Youtube [23] dépense des millions de dollars en bande passante pour mettre à disposition son service vidéo ou pour stocker les contenus vidéo [27]. On peut facilement extrapoler et estimer que cela serait encore plus onéreux pour un service vidéo de haute qualité (HDTV). En multipliant le nombre de serveurs, l'audience maximale avec des CDN dépendrait directement du nombre de serveurs déployés. Le serveur source et les serveurs du CDN connaîtraient également le problème récurrent de goulot d'étranglement au niveau des serveurs à mesure que la population augmente. De plus, avec un service de vidéo en direct, il est également très difficile de maintenir à jour tous les serveurs CDN avec le serveur source. Les CDNS ont ainsi un coût important d'infrastructure, ce qui limite leur utilisation pour

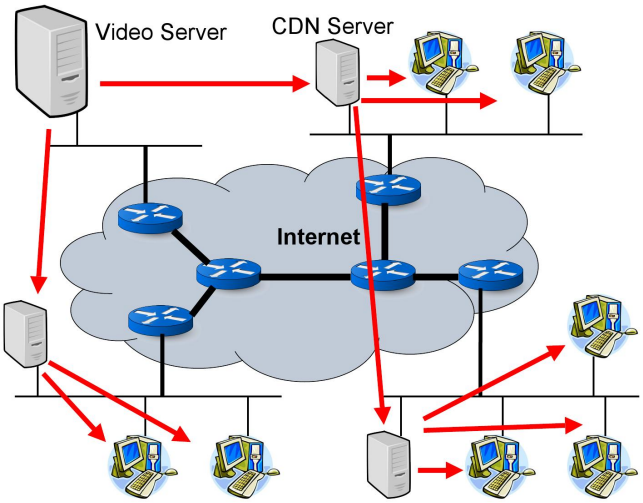


FIG. B.2 – CDN serveurs

les fournisseurs de contenu. Finalement, une solution basée sur les CDN est limitée en terme de passage à l'échelle et est extrêmement coûteuse en infrastructure et bande passante.

Les services de diffusion de vidéo en direct pourraient alors bénéficier des fonctionnalités de communications de groupe. Toutefois, l'architecture de l'Internet a été conçu pour les communications unicast (un-vers-un) et pas pour les communications multicast (un-vers-plusieurs). Le multicast de niveau réseau (IP-multicast) a dont été proposé pour ajouter des fonctionnalités de communication de groupe à l'Internet [28]. Dans ce cas, les routeurs de l'Internet sont alors responsables de la duplication des paquets vers chaque destination et ne diffusent seulement qu'une copie du paquet par lien, ce qui réduit le trafic vers toutes les destinations. Cette solution est illustrée sur la figure B.3.

Malheureusement, le déploiement du multicast réseau reste très limité à l'échelle de l'Internet pour plusieurs raisons complexes qui ne sont pas uniquement dues à des problèmes d'ordre technique, mais à des soucis d'ordre économique voire politique [29]. Pour ne citer que certains d'entre eux, le multicast réseau nécessite des changements profonds des infrastructures réseaux (routeurs) ce qui ralentit fortement son déploiement. Ces changements présentent des coûts d'infrastructures importants pour les fournisseurs d'accès à l'Internet (FAI) afin que leurs routeurs supportent les fonctionnalités de communication de groupe. Le multicast de niveau réseau présente également de nombreux problèmes tels que le trafic entre les FAI ou la sécurité (DDoS). Enfin, il n'y a pas de modèle économique pour les FAI pour transporter le trafic multicast et modifier leur infrastructure dans ce but. Ainsi, le manque de déploiement du multicast réseau limite son utilisation pour offrir un service de diffusion de télévision à l'Internet.

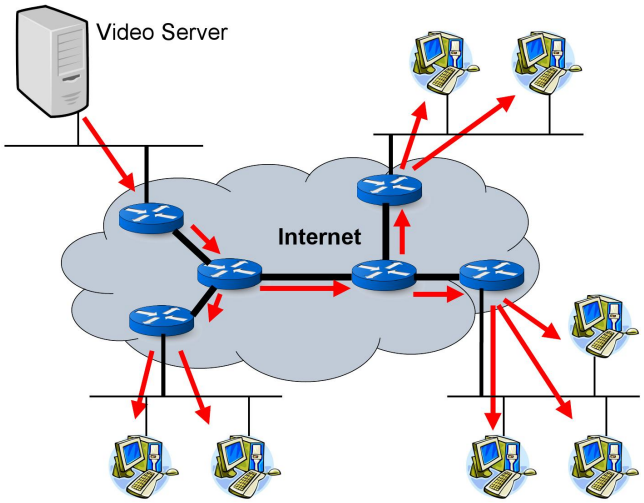


FIG. B.3 – Multicast

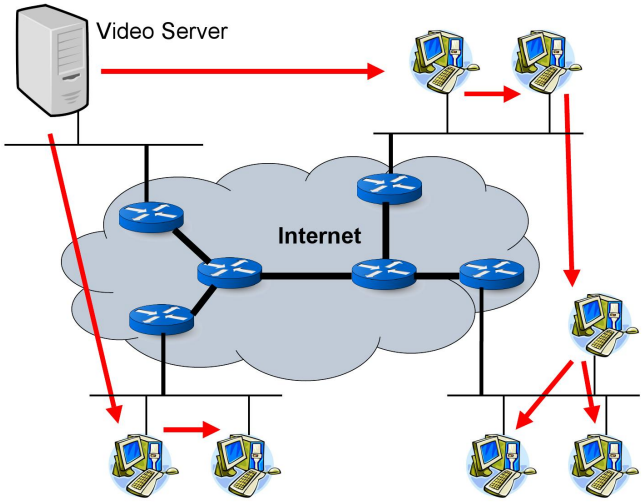


FIG. B.4 – Pair-à-Pair

Ces dernières années, après le succès de la technologie P2P pour les applications de partage de fichiers [21,22,30] ou de téléphonie [31], le modèle pair-à-pair a été pressenti pour diffuser des flots vidéo en direct sur l'Internet. il y a eu un intérêt significatif à utiliser le modèle pair-à-pair pour la diffusion de vidéo en direct sur l'Internet. Le concept clé du P2P est d'encourager les participants à être à la fois clients et serveurs. La technologie pair-à-pair ne requiert pas le soutient des routeurs de l'Internet ou de l'infrastructure réseau. Par conséquent, cette solution est très peu coûteuse et facile à

déployer. Dans les réseaux P2P, la source du contenu n'est pas la seule à transmettre les flots vidéos et à partager ses ressources (CPU, bande passante). Tous les participants, dans notre cas les utilisateurs terminaux (les *pairs*), téléchargent la vidéo mais la transmettent également à d'autres pairs. Tous les utilisateurs partagent donc leurs ressources et aident les autres pairs à obtenir les données. Cela est illustré sur la figure B.4. Ainsi, les ressources disponibles pour diffuser le contenu augmente avec le nombre d'utilisateurs, ce qui permet d'atteindre un très grand nombre d'utilisateurs et donc une très large audience.

Il y a eu de nombreuses propositions d'architectures P2P de diffusion de vidéo en direct. Ces architectures de streaming en direct peuvent être classifiées de façon générale en deux grande catégories : celles qui sont basées sur la construction d'arbre (*tree-based*) et celles qui sont basées sur la construction de maillage (*mesh-based*). Pour la construction d'arbre, nous pouvons aussi distinguer la construction d'un seul arbre (*single-tree*) et celle d'arbres multiples (*multiple-trees*). En fait, cette classification suit l'ordre chronologique des apparitions de ces architectures. L'approche *single-tree* est la première à avoir été proposée et fournit seulement des fonctionnalités de communication de groupe pour diffuser la vidéo en direct à un groupe de pairs. Ensuite, l'approche *multiple-trees* a été introduite pour pallier les défauts de l'approche *single-tree*. Récemment, l'approche basée sur les maillages est apparue et est actuellement l'architecture de référence pour concevoir des applications P2P-TV. Cette classification met en évidence l'évolution des propositions et présente les problèmes auxquels les concepteurs d'architectures P2P-TV sont confrontés pour définir des architectures efficaces.

Pour ce résumé en français, nous allons détailler le fonctionnement des différentes approches architecturales. Le fonctionnement des architectures P2P-TV est étudié dans la version longue de cette thèse. Toutefois, nous résumons les architectures étudiées en les présentant dans un tableau récapitulatif.

B.2.1 Architectures construisant un arbre

Comme expliqué précédemment, le multicast de niveau réseau a été proposé pour ajouter des fonctionnalités de communication de groupe à l'Internet mais n'a pas été largement déployé. Dans une communication multipoints de niveau réseau, la source vidéo est connectée à tous les participants à travers un arbre de diffusion formé par les routeurs du réseau. Pour pallier le manque de déploiement au niveau réseau, l'idée fut alors d'implémenter ces fonctionnalités de communication de groupe en bordure du réseau, c'est à dire au niveau applicatif. La source vidéo et les récepteurs forment alors un réseau de recouvrement au niveau applicatif (application-level overlay network) pour distribuer le contenu vidéo.

B.2.1.1 Approche à arbre unique

Pour les architectures construisant un arbre unique (*single-tree*), comme pour le multicast, le plan de livraison des données est un arbre construit parmi tous les récepteurs et dont la racine est la source de la vidéo. Chaque paquet vidéo est alors disséminé sur

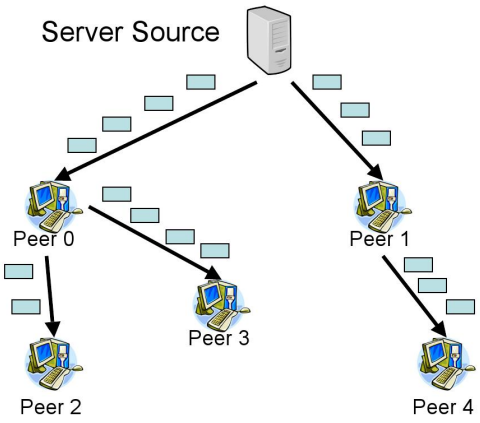


FIG. B.5 – Approche *Single-tree* : le plan de donnée est un arbre de diffusion entre tous les pairs.

la même structure d'arbre construite entre les pairs. Un plan de contrôle est utilisé pour gérer l'appartenance au groupe et pour aider les nouveaux pairs à rejoindre le plan de données afin de recevoir le contenu. L'approche *single-tree* a une structure d'overlay très organisé et les noeuds dans cet arbre ont des relations *parent/enfant* clairement définies. Les pairs distribuent la vidéo en transmettant les données à leurs enfants.

Cette approche est illustrée sur la figure B.5. Parmi les exemples d'architectures les plus représentatifs, citons Peercast, NICE, Narada et Bullet [32, 34, 36, 37] qui sont brièvement présentés dans le tableau B.1 mais largement étudiés dans la version longue de cette thèse. Chacune de ces architectures utilise un plan de contrôle différent pour gérer l'appartenance au groupe (arbre, cluster, maillage), au dessus duquel repose le plan de données qui est un arbre de diffusion parmi tous les récepteurs et dont la racine est la source vidéo.

B.2.1.2 Approche à arbres multiples

Les architectures basées sur une approche à arbre unique (single-tree) ont été les premières propositions pour le streaming multimédia en direct à travers des réseaux P2P. Au niveau applicatif, elles organisent rigoureusement les pairs en un arbre de diffusion comme cela était le cas pour le multicast de niveau réseau. L'approche single-tree présente cependant deux principaux défauts : (i) une grande majorité des noeuds se retrouvent être des noeuds feuilles et leurs ressources ne sont pas utilisées. (ii) les pannes de noeuds des niveaux supérieurs de l'arbre perturbent la dissémination des données dans toute la structure de l'overlay.

Lorsque les noeuds sont des feuilles de l'arbre de diffusion, ils ne reçoivent que les données et ne les retransmettent à aucun autre pair puisqu'ils n'ont pas d'enfant dans

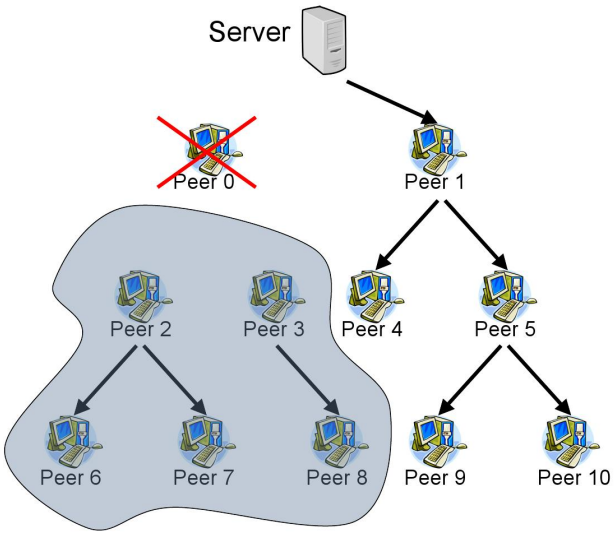


FIG. B.6 – Conséquences du départ d'un pair dans une approche à arbre unique.

la structure d'arbre. Si une grande majorité des noeuds sont feuilles, alors une grande partie des ressources de ces pairs n'est pas utilisée et c'est un problème important pour le passage à l'échelle de cette approche.

Les départs ou autres pannes des noeuds, et particulièrement ceux situés proche de la source, peuvent perturber la dissémination des données à un grand nombre de descendants dans l'arbre. Nous avons illustré ce scénario sur la figure B.6. Ce phénomène est plus généralement appelé le *churn of peers* (va-et-vient des pairs). Le *churn of peers* est l'arrivé et le départ des pairs dans le réseau à n'importe quel moment.

Dans le cas des architectures P2P que nous étudions, les pairs sont des hôtes terminaux qui ne sont pas entités conçues pour rester éternellement dans le réseau contrairement aux routeurs de l'Internet. Les hôtes terminaux sont par contre sujets à des pannes et le comportement de leurs utilisateurs peut avoir un impact sur la durée de vie des pairs dans le réseau. Dans le cas de pannes, le départ d'un pair va interrompre momentanément la diffusion de la vidéo à tous les pairs situés dans le sous arbre dont la racine est le pair défaillant. Tous ces pairs vont ainsi subir des pertes de paquets vidéos. Le *churn of peers* est un problème très important à prendre en compte dans les réseaux P2P. C'est d'autant plus le cas pour la diffusion de flots vidéos en direct en raison de la nature continue du contenu vidéo et de sa sensibilité aux délais de traversée du réseau.

Afin de pallier les défauts de l'approche à arbre unique, des structures plus résistantes comme les arbres multiples ont été définies. Dans ce cas, la source encode le flot vidéos

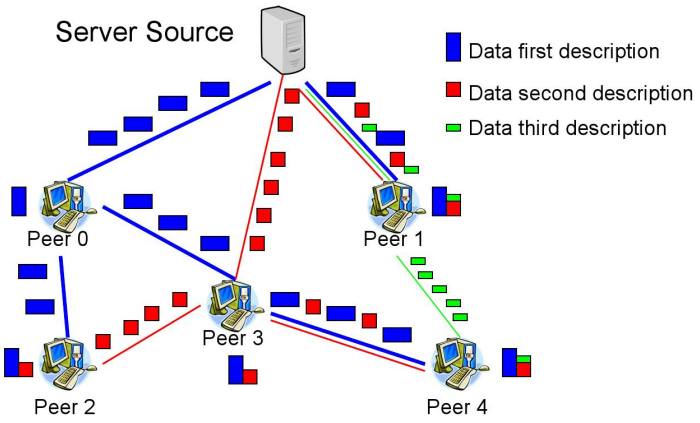


FIG. B.7 – Approche multi-arbres : le plan de données est composé de plusieurs arbres de diffusion entre les pairs.

en plusieurs sous-couches de moins bonnes qualités (*sub-streams*) en utilisant les techniques d'encodage en couches [38] (LC) ou à multiples descriptions [39] (MDC). La source distribue chaque sous-couche le long d'un arbre de diffusion spécifique à cette sous-couche (figure B.7). La qualité reçue par un récepteur dépend du nombre de sous-couches qu'il a reçues. Un pair peut choisir de recevoir les données de tous les arbres de diffusion et obtenir la qualité vidéo maximale, ou bien ne sélectionner que certaines sous-couches et adhérer à un sous-ensemble des arbres de diffusion et adapter la qualité vidéo reçue à ses propres besoins et capacités. Cette technique permet également aux pairs avec de faibles capacités de transmission (*upload*) de ne transmettre que quelques couches et pas la vidéo en entier. Ces architectures s'adaptent aux capacités hétérogènes des pairs du réseau.

Il existe des avantages déterminants de l'approche construisant des arbres multiples sur celle avec un unique arbre. Avec plusieurs arbres de diffusion, les ressources de tous les noeuds peuvent être utilisées tant que les noeuds ne sont pas feuilles dans au moins un des arbres. Un pair qui est feuille dans un des arbres devrait alors être un noeud interne dans un autre arbre pour compenser et partager ses ressources avec les autres pairs. Dans le cas du départ d'un pair, la résistance totale du système P2P se trouve améliorée puisque les pairs enfants n'ont plus leur flot vidéo complètement interrompu par la panne d'un des pairs se situant en amont et continuent à recevoir des données via les autres arbres de diffusion.

Des exemples d'architectures construisant des arbres multiples sont présentés dans le tableau B.1.

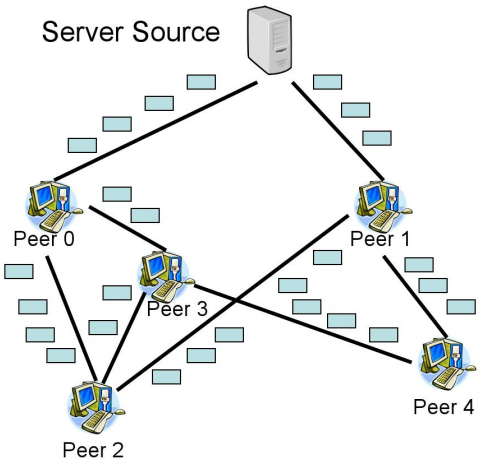


FIG. B.8 – Architectures construisant un maillage.

B.2.2 Architectures construisant un maillage

Les architectures basées sur la construction d'arbre organise les pairs en des topologies virtuelles très statiques, où les pairs ont des relations très définies et obtiennent les données de leur parent (arbre unique) ou de plusieurs parents (arbres multiples). Les départs incessants des pairs (*churn*) peuvent interrompre la livraison des données pour un grand nombre de pairs organisés en une structure d'arbre unique. Même si les architectures à arbres multiples ont été introduites pour résoudre ce problème, la gestion de plusieurs arbres avec un environnement aussi dynamique est très complexe.

L'approche construisant un maillage a été introduite récemment grâce à BitTorrent [21], un système P2P de partage de fichiers qui a été déployé avec succès sur l'Internet. À l'inverse des architectures construisant des arbres, les architectures construisant des maillages ne construisent pas une topologie virtuelle régulière pour diffuser les données aux utilisateurs. Il n'y a pas de topologie statique de créée et les pairs du réseau établissent des relations de pairs dynamiquement en fonction de la disponibilité des données. Au lieu de réparer sans cesse une structure statique dans un environnement P2P hautement dynamique, la disponibilité des données parmi les pairs guide leurs relations. Les architectures basées sur la construction de maillage sont aussi souvent appelées s les approches orientés-données.

De la même façon qu'avec BitTorrent, les pairs forment un maillage aléatoire dynamique entre eux et maintiennent des relations avec de nombreux pairs en même temps. Ils envoient (ou reçoivent) du contenu vers (ou de) plusieurs voisins en fonction de la disponibilité des données. Dans le cas de départ d'un noeud voisin, le pair continue à recevoir les données des autres voisins qu'il connaît. Pendant toute leur session vidéo, les pairs continuent à découvrir de nouveaux pairs et à établir de nouvelles relations de pairs. Puisque chaque pair maintien son propre voisinage dynamiquement,

l'approche en maillage est très robuste au va-et-vient des pairs *churn of peers*. En effet, les pairs connaissent immédiatement de nouveaux pairs de substitutions. Globalement, des études ont montré que les architectures en maillage étaient plus performantes que celles qui construisent des arbres [48–51].

Pour les approches en arbre, le flot vidéo est transmis de la source à tous les pairs en suivant la structure d'arbre construite (ou plusieurs arbres). Dans les approches en maillage, les différentes parties de la vidéo peuvent être reçues par différents voisins, rendant obsolète le concept de flot vidéo. Comme pour BitTorrent, la vidéo est divisée en petits morceaux appelés *chunks*. Chaque morceau est diffusé à tous les pairs à travers le maillage. Puisque tous les chunks peuvent prendre différents chemins pour atteindre leur destination, ils peuvent arriver dans n'importe quel ordre. Un pair va donc devoir conserver (*bufférisér*) les chunks reçus pour les rejouer dans le bon ordre de lecture.

Dans l'approche construisant un maillage, on peut distinguer deux politiques bien distinctes d'échanges de chunks entre les pairs. D'un côté il y a l'envoi simple d'un chunk, aussi appelé *push policy*, dans lequel chaque pair envoie activement le nouveau chunk reçu à ses voisins. Seulement, un pair pourrait malencontreusement envoyer un chunk à un voisin ayant déjà reçu le même chunk, ce qui gaspillerait les ressources en bande passantes du réseau. D'un autre côté, il existe la politique d'envoi d'un chunk lorsque celui-ci est explicitement demandé par un pair. Cette politique est appelée *pull policy*. Avec cette politique d'envoi de chunks, chaque pair demande explicitement un chunk à l'un de ses voisins. Pour ce faire, les pairs envoient périodiquement à leurs voisins des informations sur les données dont ils disposent en transmettant des *buffer maps*. Typiquement, le *buffer map* est un vecteur de bits où la présence d'un chunk est indiquée par un bit à 1, et son absence par un bit à 0 [52]. Après avoir obtenu les *buffer maps* de ses voisins, un pair peut demander explicitement les chunks à ses voisins et organiser son téléchargement. Cette dernière politique d'envoi de chunks (*pull policy*) réduit le nombre de transmissions multiples du même chunk puisqu'un pair n'obtient un chunk qu'après en avoir fait la demande explicite à un voisin. Cependant, cette politique ajoute de nouveaux messages de signalisation et des délais additionnels pour récupérer les chunks. En effet, il faut alors au moins un délai d'aller-retour (RTT) entre une demande de chunk et son envoi pour l'obtenir.

Les approches construisant un maillage pour diffuser les flots multimédias sont présentées dans le tableau B.1.

B.2.3 Discussion

DONet est le tout premier exemple d'architecture construisant un maillage entre les pairs et la plupart des autres approches ne firent que l'améliorer légèrement en définissant de nouvelles fonctionnalités comme des mécanismes d'incitation (Pulse) ou des mécanismes de *network coding* (R^2).

Seuls DONet (Coolstreaming) et GridMedia ont été récemment implémentés et déployés sur l'Internet. Ces deux systèmes affirment avoir compté entre 80,000 et 220,000 utilisateurs en même temps [58, 69–71], ce qui ne démontre qu'approximativement la faisabilité de ce type d'architecture.

Tab. B.1 – Récapitulatif des Architectures

Nom	Plan de Contrôle	Observations
Arbre unique		
PeerCast	arbre	inspiré du multicast IP
Nice/ZigZag	clusters	ZigZag équilibre la charge des clusters
Narada	maillage	réseaux de petites échelles
Bullet	maillage/arbre	vers les arbres multiples
Arbres multiples		
Splitstream	DHT	arbres disjoints
CoopNet	serveur centralisé	encodage MDC
Pals	-	encodage à base principale (LC)
Maillage		
DONet	algorithme épidémique	*pull policy*
GridMedia	algorithme épidémique	*push-pull policy* (hybride)
Pulse	algorithme épidémique	mécanismes incitatifs
R^2	algorithme épidémique	*network coding* et *push policy*
Prime	centralisé	*push-pull policy* et MDC
Chunkyspread	marche aléatoire	arbre par paquets vs. arbre par flots

Pour mettre ce résultat en perspective, TF1, la chaîne de télévision la plus populaire en France, a une audience moyenne qui oscille entre 5 et 10 millions de téléspectateurs en même temps [72]. En France, pour certains évènements comme des matchs de football, le pic d'audience peut même atteindre 20 millions de téléspectateurs en même temps, soit environ un tiers de la population Française. Certains évènements planétaires comme la coupe du monde de football, les Jeux Olympiques ou la finale du championnat de football américain (Superbowl) peuvent compter plusieurs centaines de millions de spectateurs en même temps. Bien sûr, il s'agit là des chiffres cumulés pour toutes les chaînes de télévision retransmettant l'évènement et pas une seule chaîne.

Cependant, les applications P2P sont utilisées mondialement et peuvent cibler une très large audience. Un scénario pourrait être celui des Jeux Olympiques, où une source diffuse en direct la finale du 100 m en athlétisme à des millions de spectateurs situés n'importe où dans le monde. Ainsi, les déploiements et les expérimentations des architectures P2P-TV comme DONet ou GridMedia sont finalement très limités et il n'est pas évident de savoir si ces architectures pourraient supporter une diffusion à une audience bien plus importante (un ou deux ordre de magnitude plus important).

En parallèle, beaucoup d'applications P2P-TV ont été déployées sur l'Internet et affirment s'inspirer directement de BitTorrent et des approches basées sur la construction de maillage. Ces applications comptent déjà des millions de téléchargements et d'utilisateurs enregistrés [73]. L'application la plus populaire est très certainement PPLive [14] mais beaucoup d'autres sont très utilisées comme PPSTream, SOPCast,

TVAnts, UUSee, TVUPlayer ou PPMate ([15–20]). Ces applications commerciales sont propriétaires et leurs codes sources ne sont pas disponibles : il n'y a aucun détail concernant l'implémentation ou l'architecture utilisée par ces applications commerciales. Le fonctionnement de ces applications est totalement inconnu alors que l'on s'attend à ce qu'elles soient massivement utilisées dans un futur proche. Il est donc très important d' étudier ces applications P2P-TV commerciales afin de comprendre leur fonctionnement général.

Dans cette thèse, nous allons étudier en détail les applications P2P-TV commerciales. Cette étude nous permettra dé découvrir les propriétés intéressantes de ces applications P2P-TV et les caractéristiques de leurs utilisateurs. Cela pourrait également être utile pour concevoir de nouveaux mécanismes ou paradigmes pour la diffusion de flots multimédia en direct via un réseau P2P.

B.3 Expériences de mesures P2P-TV locales

Comme nous l'avons expliqué précédemment, au cours de nos travaux, nous nous sommes aperçus que beaucoup d'architectures P2P avaient déjà été proposées pour la diffusion de flots multimédia en direct. Seules quelques-unes d'entre elles avaient réellement été implémentées et testées mais à petites échelles. La plupart de ces propositions ont été validées via la simulation de leur architecture et de leurs mécanismes. En parallèle, des applications commerciales ont été déployées sur l'Internet sans préciser la moindre information concernant leur architecture ou implémentation [14–20]. Il n'y avait donc aucune connaissance de ces systèmes et aucune information n'était disponible pour les évaluer correctement.

Ainsi, afin de combler l'écart entre les architectures académiques et les applications commerciales, nous avons effectué d'importantes expériences de mesure de trafic des applications P2P-TV commerciales. Nous souhaitions comprendre le véritable fonctionnement de ces nouveaux systèmes.

De plus, les applications P2P-TV devenaient de plus en plus populaires. Comme le trafic P2P et le trafic vidéo constituent déjà une part importante du trafic de l'Internet, l'on s'attend à ce que les applications P2P-TV contribuent à augmenter le trafic de l'Internet et à compter pour une part importante de celui-là. Par conséquent, il est alors important d'évaluer l'impact de ce trafic sur le réseau.

Nous allons donc présenter notre expérience de mesure et présenter les principaux résultats de notre analyse de trafic P2P-TV.

B.3.1 Expériences de mesures

Afin de mieux comprendre les propriétés du trafic des applications P2P-TV, leurs mécanismes sous-jacents et le comportement de leurs utilisateurs, nous avons étudié quatre applications pendant la coupe du monde de football 2006 (FIFA). L'analyse de plusieurs applications nous permettra d'étudier ces systèmes sans être trop étroitement liés à leur implémentation, ce qui nous permettra de généraliser nos résultats.

Pour nos expériences de mesures, nous avons choisi les applications P2P-TV les plus populaires à savoir PPLive, PPStream, SOPCast et TVAnts.

B.3.1.1 Plateforme expérimentale de mesures

Nous avons collecté une importante quantité de données en mesurant la plupart des matchs de la coupe du monde de football avec différentes applications en même temps. La taille totale de notre collection de données est d'environ 250 GB. Dans ces travaux, nous nous intéressons à quatre traces, une pour chaque application, que nous avons collectées le 30 juin 2006 dans le réseau de l'université Pierre et Marie Curie - Paris 6. Pendant un match, nous avons utilisé deux ordinateurs en faisant fonctionner sur chacun l'une des applications ainsi que `WinDump` [78] pour collecter le trafic. Ainsi, nous avons récolté deux traces de trafic pour chacun des deux matchs, une par application.

Notre plateforme de mesure est décrite sur la figure B.16. Pour collecter les paquets, nous avons utilisé deux PCs équipés de processeur cadencé à 1.8GHz, de carte graphique

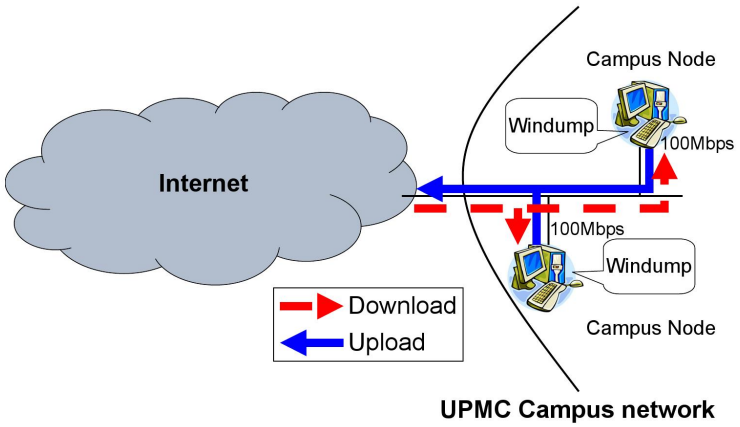


FIG. B.9 – Plateforme expérimentale de mesures. Chaque noeud est un PC standard directement connecté à l'Internet via le réseau de notre université.

TAB. B.2 – Résumé des traces.

	PPLive	**PPStream**	**SOPCast**	**TVAnts**
Durée (s)	13,321	12,375	12,198	13,358
Taille [MB]	6,339	4,121	5,475	3,992
Téléchargement[%]	14.11	20.50	16.13	24.76
TCP	14.09	20.50	0.23	14.71
UDP	0.02	$\approx$ 0.00	15.90	10.05
Émission[%]	85.89	79.50	83.87	75.24
TCP	85.81	79.50	3.89	61.67
UDP	0.08	$\approx$ 0.00	79.98	13.57

standard et fonctionnant sous *Windows XP* Les PCs étaient connectés à l'Internet à travers le réseau de notre université via une liaison Ethernet à 100Mbps.

B.3.2 Étude du trafic P2P-TV

Les applications génèrent du trafic en utilisant différents ports et protocoles. Le tableau B.2 résume les informations des traces étudiées. La durée des traces est de 225 minutes, plus longue qu'un match de football afin de pouvoir capturer les effets qui peuvent se produire pour de tels évènements en directs.

Il y a plus de trafic en upload qu'en download. Par exemple, la figure B.10 montre la quantité de téléchargement et d'émission pour TVAnts. Le débit de téléchargement est relativement constant et à un niveau proche du débit vidéo alors que le débit d'émission fluctue largement à dés débits plus élevés. Nos noeuds téléchargent la vidéo

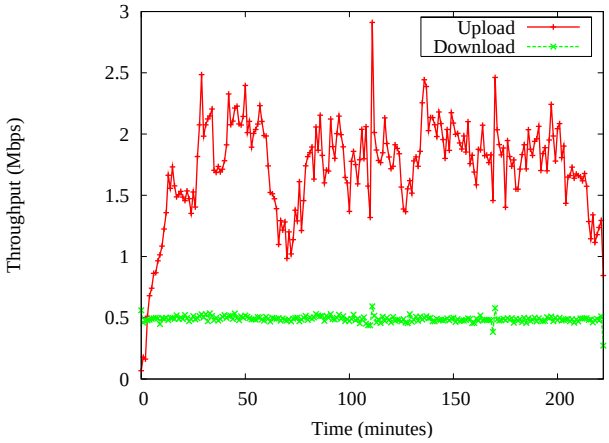


FIG. B.10 – Débits montant et descendant pour TVAnts.

à débit constant mais partagent leurs grandes capacités d'émissions avec de nombreux autres pairs.

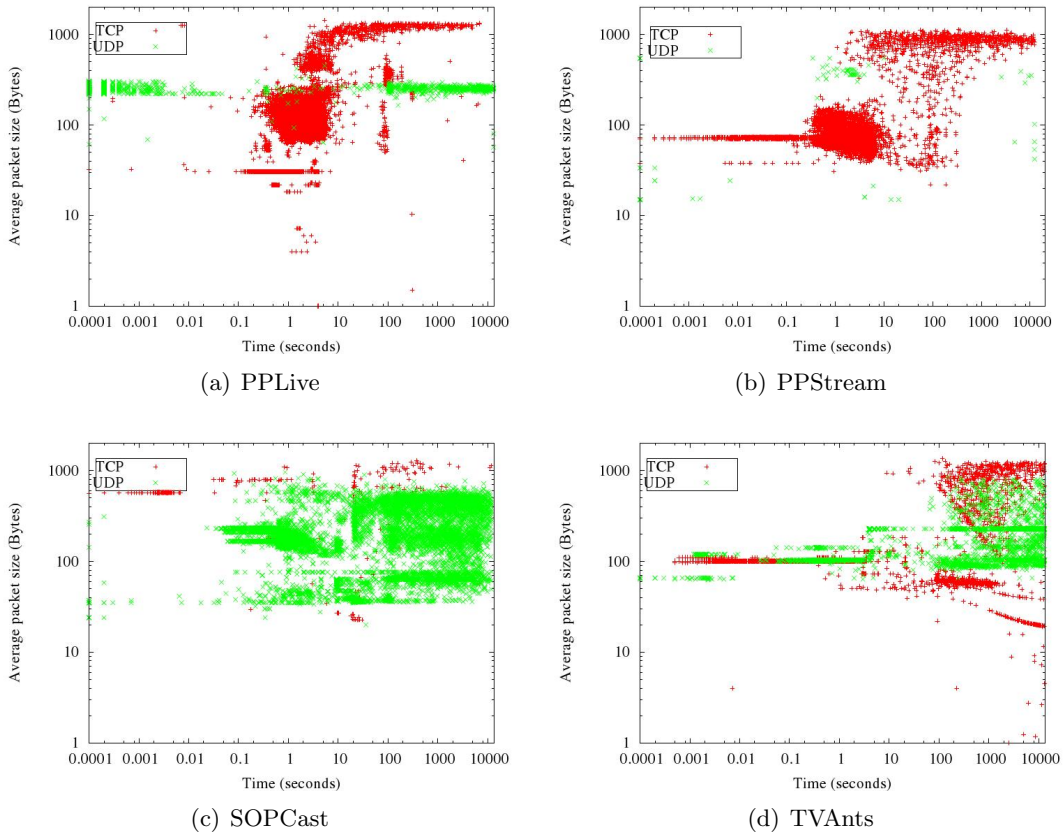
En effet nos noeuds disposent d'une connexion à l'Internet à très haut débit via un lien Ethernet 100 Mbps.

B.3.2.1 Protocoles de transport

Les applications P2P-TV utilisent du trafic de signalisation pour s'échanger les informations entre les pairs et du trafic vidéo pour s'échanger les morceaux de vidéos (chunks). Les sessions vidéos sont susceptibles d'avoir de longues durées puisque lorsqu'un pair découvre un autre pair pour échanger des données, il est préférable de maintenir cette session le plus longtemps possible pour échanger des données. Les sessions de signalisations auraient par contre plutôt tendances à être de durée plus courte car elles ne transportent que des informations pour permettre aux pairs de décider s'ils doivent ou non établir une session vidéo entre eux. De plus, les paquets vidéos sont supposés être de grandes tailles alors que ceux de signalisation peuvent être de plus petites tailles.

La figure B.11 présente la taille moyenne des paquets des sessions en fonction de la durée de celles-ci.

PPLive (figure B.11(a)) et PPStream (figure B.11(b)) ont un trafic TCP similaire et utilisent un peu UDP. Les sessions UDP de PPLive varient de durée très courtes ($< 1s$) à des durées très longues alors qu'elles sont relativement constantes pour PPSTream, mais la taille moyenne des paquets des sessions UDP est faible (≈ 250 octets). Le trafic TCP de PPLive et PPStream présente deux ensembles de points : celui situé au milieu de la figure est pour les sessions de courte durée transportant de petits paquets. Ce sont des cessions de signalisation. L'autre ensemble en haut à droite des figures est les sessions vidéo puisque ces sessions sont de longue durée et transportent des paquets de



(a) PPLive (b) PPStream

(c) SOPCast (d) TVAnts

FIG. B.11 – Taille moyenne des paquets des sessions en fonction de la durée de celles-ci.

tailles importantes. PPLive et PPSTream utilise TCP pour transporter tous les trafics vidéo et de signalisation.

SOPCast utilise quant à lui presque entièrement UDP (figure B.11(c)). On peut aussi apercevoir des ensembles de sessions au milieu (signalisation) et dans le coin supérieur droit (vidéo) mais ils ne sont pas aussi nettement définis. TVAnts semble lui bien équilibré entre le trafic TCP et UDP (figure B.11(d)). On peut distinguer deux ensembles de sessions mais chacun contient à la fois du trafic TCP et UDP. TVAnts transporte les trafics vidéo et de signalisation indifféremment sur TCP ou UDP. Cependant, TVAnts utilise plus largement TCP que UDP ($\approx$ 75%, tableau B.2).

Le tableau B.3 résume nous observations de la figure B.11 : toutes les applications transportent le trafic de façon différente. PPLive et PPSTream utilisent seulement TCP pour la vidéo et la signalisation et UDP pour une petite partie du trafic de signalisation. TVAnts utilise indifféremment TCP ou UDP pour tous types de trafic, et SOPCast utilise presque entièrement UDP.

PPSTream et PPLive utilise par défaut TCP pour transporter du contenu multimédia. À l'inverse, SOPCast utilise UDP pour le trafic vidéo. La présence de quelques

Tab. B.3 – Récapitulatif du trafic observé.

	PPLive	**PPStream**	**SOPCast**	**TVAnts**
Trafic vidéo	TCP	TCP	UDP un peu TCP	TCP UDP
Trafic de signalisation	TCP un peu UDP	TCP	UDP un peu TCP	TCP UDP

sessions TCP indique toutefois que SOPCast est aussi capable d'utiliser un autre protocole de transport lorsque cela est nécessaire pour joindre les utilisateurs distants sur l'Internet. D'ailleurs, lors d'une autre expérience, nous avons modifié les règles de notre pare-feu (*firewall*) en lui demandant de rejeter tous les paquets UDP. Nous avons pu alors constaté que SOPCast était capable de n'utiliser que TCP pour les trafics vidéo et de signalisation. Cela corroborait le fait que SOPCast utilise UDP comme choix par défaut mais peut aussi utiliser TCP pour atteindre tous les hôtes de l'Internet. En fonction de l'accessibilité des hôtes distants, TVAnts est capable de s'adapter et d'utiliser chacun des protocoles de transports. Si l'on considère le trafic de toutes les applications en même temps, alors le trafic vidéo est majoritairement transporté par TCP, alors que ce n'est pas le plus adapté pour les applications multimédia en temps-réel. Comme pour toutes les applications réseau, TCP a des avantages comme sa fiabilité mais également ses inconvénients comme un surcoût en trafic de signalisation ou un temps de traitement important. La fiabilité de TCP et les retransmissions que cela occasionne peut également s'avérer être un défaut si cela prend trop de temps. En effet, si le paquet arrive après son instant de lecture alors la retransmission aura été inutile et aura gaspillé de la bande passante. Nous pensons que le choix de TCP pour transporter les données a été décidé par les concepteurs pour permettre aux applications d'atteindre tous types d'utilisateurs de l'Internet, même si ils sont situés derrière des systèmes de translations d'adresses [82] (NAT).

B.3.2.2 Ports

Le tableau B.4 montre les ports utilisés par ces applications. PPLive et SOPCast ont un comportement identique puisque notre machine utilise le même port pour toutes les communications avec les pairs, alors que ceux-là utilisent un large spectre de ports différents. PPStream a le même comportement excepté qu'il utilise un même port distant et trois ports locaux pour UDP. Il est également très intéressant de constater que PPSTream et PPLive utilise le port local UDP 5747. Un comportement particulier est cependant observé pour TVAnts, qui utilise le port 16800 localement et à distance pour UDP et TCP. TVAnts configure probablement un port par défaut à chaque nouvelle installation et qui peut être changé par l'utilisateur ultérieurement. En regardant le tableau B.4, il est aussi clair que l'on ne peut pas identifier le trafic P2P-TV en regardant uniquement les ports des protocoles de transports.

Tab. B.4 – Port utilisés (pourcentage de paquets).

		PPLive	**PPStream**	**SOPCast**	**TVAnts**
Pairs Distants	TCP	plusieurs	plusieurs	plusieurs	16800 (> 25%)
	UDP	plusieurs	7201 (100%)	plusieurs	16800 (> 60%)
Pairs contrôlés	TCP	10549 (> 99%)	11430 (> 99%)	8516 (> 99%)	16800 (> 71%)
	UDP	5747 (100%)	5747 (42%), 11430 (54%), 65535 (4%)	8516 (> 99%)	16800 (> 99%)

Tab. B.5 – Proportion de trafic de signalisation.

	PPLive	**PPStream**	**SOPCast**	**TVAnts**
Total	4.1%	13.6%	19.3%	10.2%
Émission	2.2%	10.8%	13.6%	7.8%
Téléchargement	19.2%	25.8%	48.5%	18.0%

B.3.3 Trafic de signalisation et vidéo

Les applications P2P génèrent du trafic vidéo et de signalisation. Le trafic de signalisation est supposé ne pas être sensible aux délais car il ne transporte pas de données ni de commandes interactives comme pour la VoD. Les trafics vidéo et de signalisation n'ont pas les mêmes propriétés comme la taille des paquets ou leurs contraintes temporelles et ils vont donc avoir un impact différent sur le réseau. Ainsi nous souhaitons séparer le trafic vidéo et de signalisation afin d'analyser leurs propriétés particulières.

Une heuristique simple pour séparer le trafic de signalisation du trafic vidéo a été proposée dans les travaux de Hei [84]. Cette heuristique fonctionne de la façon suivante : si une session compte au moins 10 gros paquets d'une taille supérieure a 1000 octets, alors cette session est considérée comme étant une session vidéo.

Le tableau B.5 montre des statistiques sur la proportion de trafic de signalisation en fonction du trafic total de toutes les applications. SOPCast produit énormément de trafic de signalisation alors que PPLive en produit le moins. La proportion de trafic de signalisation envoyée est bien moindre que celle reçue.

B.3.4 Comportement d'échelle du trafic P2P-TV

Nous souhaitons analyser le trafic collecté à différentes échelles temporelles. Pour cela nous avons calculé le spectre d'énergie du trafic en utilisant les méthodes de transformées en ondelettes communément utilisées dans ces travaux [85]. Le plus petit intervalle de temps que nous considérons est de 20 millisecondes. Nous n'avons pas considéré les paquets sans données comme les acquittements TCP.

Nous analysons le trafic en utilisant l'outil LDestimate [86], qui produit un graphique logarithmique du spectre d'énergie du trafic.

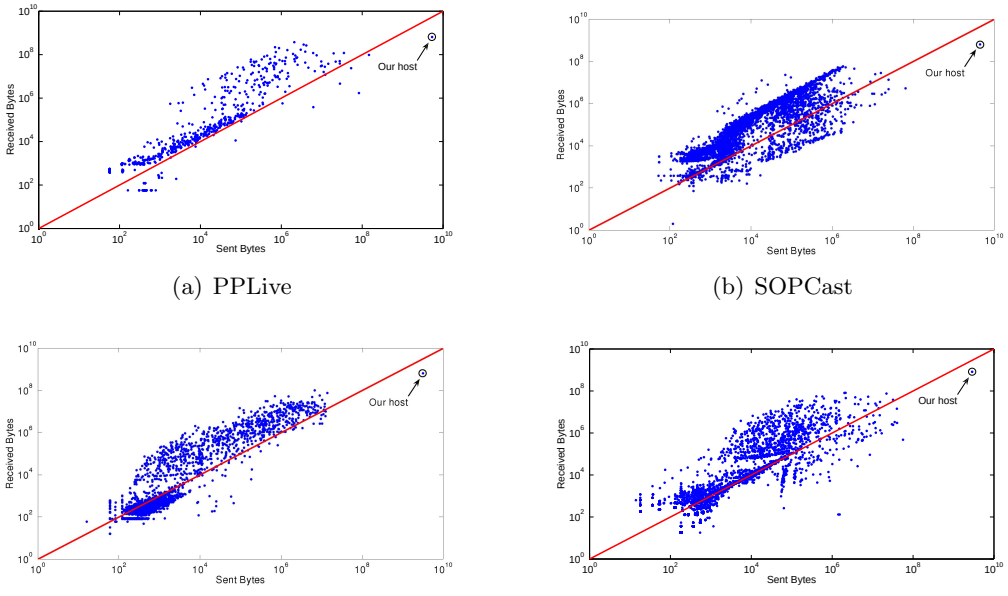
Pour chaque application nous avons séparé le trafic montant et descendant, ainsi que le trafic vidéo et le trafic global en utilisant l'heuristique de filtrage présentée précédemment.

B.3.4.1 Différences entre trafic TCP et UDP

Pour les applications utilisant principalement TCP3.7-3.10, les spectres d'énergie du trafic montant sont similaires, alors que les trafics descendant de vidéo et de signalisation se ressemblent seulement jusqu'à $j = 9$. Le spectre d'énergie du trafic montant est quant à lui clairement différent du trafic descendant. Cependant, les spectres d'énergie des applications TCP se ressemblent entre eux. Pour SOPCast (figure 3.9) , les quatre spectres d'énergie se ressemblent tous, mais ils sont complètement différents de ceux des applications TCP. En particulier, les applications TCP présent une bosse dans le spectre d'énergie à l'octave $j = 8$ (5.12s). Une telle bosse est plus prononcée en upload qu'en download et témoigne d'un phénomène périodique à ces intervalles de temps. Les mécanismes de TCP pourraient créer des phénomènes périodiques mais pas pour ce long intervalle de temps. Si le comportement périodique provenait de la diffusion même de la vidéo, alors SOPCast aurait aussi présenté ce type de phénomène périodique. Nous ne sommes toujours pas en mesure de statuer sur ce phénomène mais cela illustre bien comment un simple choix de conception peut avoir un impact important sur les propriétés du trafic généré. Pour SOPCast, nous observons une croissance linéaire de ces spectres pour tous les types de trafic. Cette croissance linéaire témoigne de la dépendance à long terme du trafic de SOPCast (LRD). Cela signifie que le trafic fluctue largement et qu'il n'est pas prédictible. Il devient alors difficile de fournir des paramètres de QoS comme une garantie des délais aux utilisateurs car les conditions du réseau évoluent sans cesse [80]. Les choix de conceptions même des applications P2P-TV vont avoir un impact immédiat sur le trafic généré. De plus, ces observations mettent en évidence le choix non trivial des protocoles de transports pour des applications multimédia à fortes contraintes temporelles. En effet, généralement les transferts de données non élastiques comme la vidéo devraient utiliser UDP mais celui-ci présente de la LRD. Cela affectera les conditions du réseau et la qualité même des flots reçus.

B.3.4.2 Impact du trafic de signalisation

Pour toutes les applications, le spectre d'énergie du trafic vidéo montant ressemble exactement à celui du trafic total. Retirer le trafic de signalisation n'a pas eu d'impact sur les propriétés du trafic montant. À l'inverse, pour le trafic descendant, les spectres d'énergie ne correspondent pas et le retrait du trafic de signalisation modifie les propriétés du spectre d'énergie du trafic descendant. Le trafic de signalisation n'a dont pas le même impact sur le trafic descendant et le trafic montant. Cette observation est importante car le trafic de signalisation est essentiel pour coordonner les échanges de données dans les systèmes P2P. D'ailleurs la proportion de trafic de signalisation est



(a) PPLive (b) SOPCast

(c) PPStream (d) TVAnts

FIG. B.12 – Octets envoyés vs. Octets reçus pour chaque pair.

plus importante pour le trafic descendant que montant. En effet, nos noeuds ont de grandes capacités et envoie la vidéo plusieurs fois alors qu'ils ne la téléchargent bien sûr qu'une fois. La proportion de signalisation en upload est donc faible par rapport au trafic vidéo. Le trafic de signalisation téléchargé provient des autres pairs qui souhaitent coordonner leur téléchargement. Ce trafic compte pour une part importante du trafic téléchargé.

L'impact significatif du trafic de signalisation sur le trafic descendant indique que les deux sens de trafic n'ont pas les mêmes propriétés ni le même impact sur le réseau. Ces observations sont importantes si l'on souhaite concevoir des modèles de trafic synthétique pour ce type d'application.

B.3.5 Comportement des pairs P2P-TV

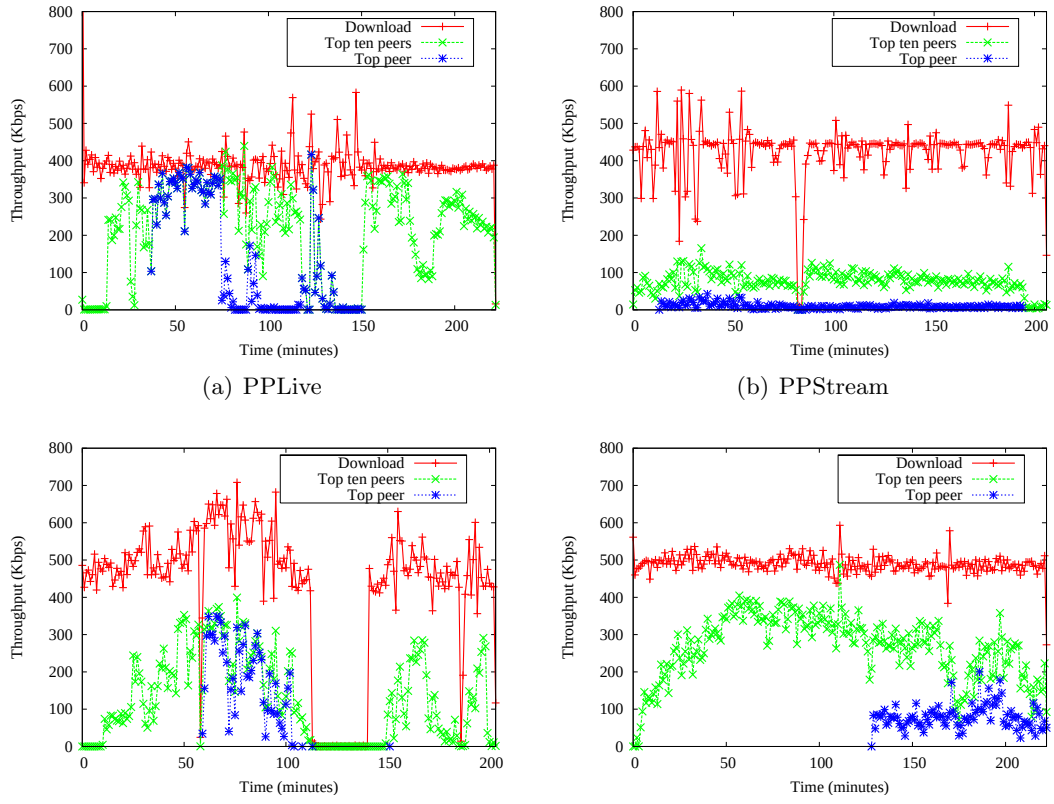
Nous souhaitons comprendre le comportement des pairs des applications P2P-TV et à la fois mettre en évidence les points communs et différences parmi les applications mesurées.

B.3.5.1 Trafic généré et reçu par les pairs

Nous nous intéressons au point de vue d'un pair et considérons tout le trafic échangé entre les pairs que nous contrôlons et tous les autres. La figure B.12 montre que les pairs distants reçoivent plus qu'ils nous envoie. Ce comportement est particulièrement pro-

TAB. B.6 – Nombre de pairs interagissant avec nos noeuds. Le nombre total et le pourcentage dans *"la région non coopérative"* *((00,0)-(10^4,10^4)).*

	PPLive	**PPStream**	**SOPCast**	**TVAnts**
Total	649	5,956	3,876	5,394
Région non coopérative	305	3,186	3,012	4,075
Pourcentage	47%	53,5%	77,7%	75,5%



(a) PPLive (b) PPStream

(c) SOPCast (d) TVAnts

FIG. B.13 – Politiques de téléchargement de la vidéo.

noncé pour PPLive et PPStream. De plus, une très grande quantité de pairs est située dans le coin inférieur gauche des figures, indiquant que leurs échanges avec nos pairs sont très faibles. Le tableauB.6 inique que 50% à 75% des pairs rencontrés n'échangent que très faiblement avec les notre.

Afin de comprendre les politiques de téléchargement, nous avons calculé la quantité

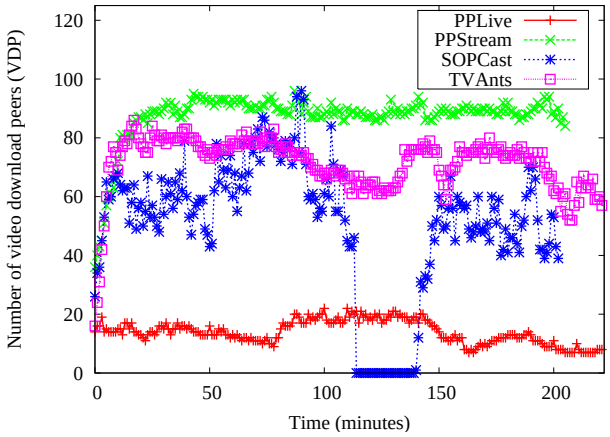


FIG. B.14 – Voisinage des pairs pour toutes les applications.

de données téléchargées par chacun de nos noeuds. Nous avons alors isolé sur la figure B.13 les dix plus gros fournisseurs de données ainsi que le plus gros fournisseurs de données.

Les politiques de téléchargement pour chacune des applications sont différentes. Certaines attendent que les pairs restent longtemps dans le réseau (PPSTream) ou peu de temps (PPLive, SOPCast) ou requièrent d'avoir une connexion haut débit pour envoyer la vidéo (PPLive) ou s'accommode de débits modérés (PPSTrean et TVAnts).

Selon l'application, un pair peut télécharger la vidéo d'un seul ou de plusieurs pairs en même temps, et la durée des sessions vidéos peut être diverses et variées. Des politiques de téléchargement différentes impliquent différentes politiques pour établir et maintenir des connexions avec d'autres pairs.

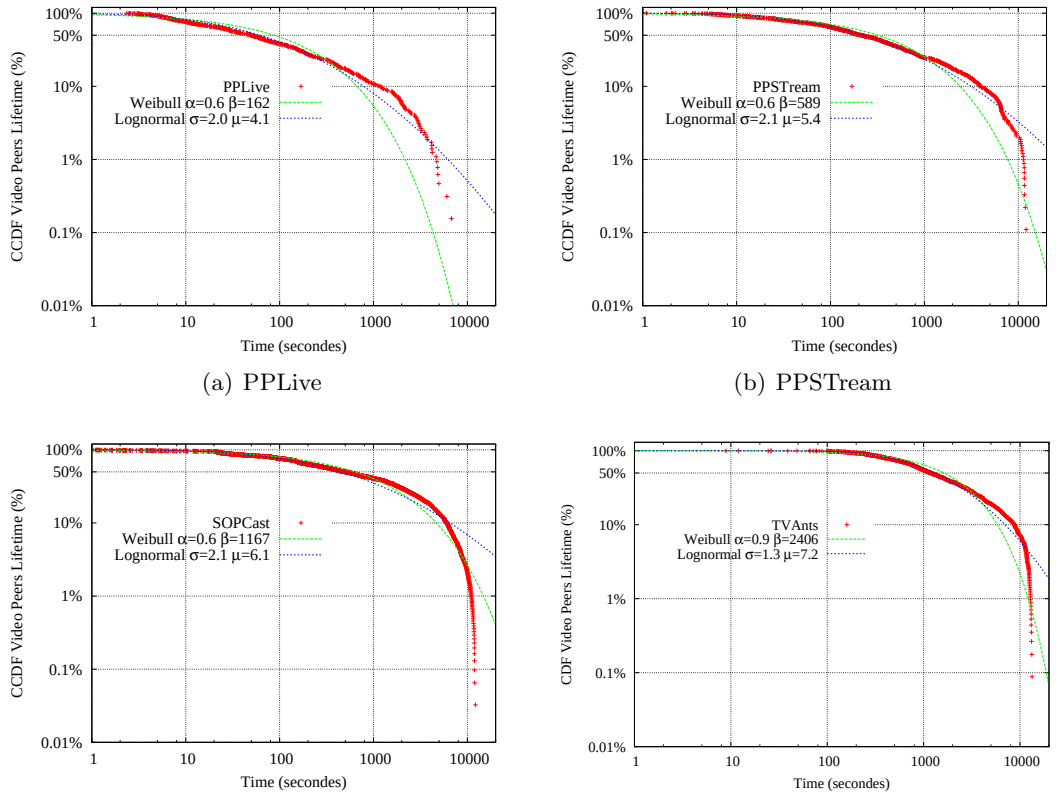
B.3.5.2 Voisinage des pairs et durée des sessions

La figure B.14 présente pour chaque application le voisinage de pairs maintenus pour télécharger la vidéo. Toutes les applications maintiennent un voisinage différents, ce qui confirme le fait que les applications utilisent différentes politiques de téléchargement.

Dans les systèmes P2P, les pairs sont responsables de retransmettre les flots aux autres. Or, les pairs ne sont pas des entités supposer rester dans le réseau indéfiniment. Il peut y avoir un va-et-vient incessant des pairs (*churn*), ce qui occasionnerait des pertes ou allongerait le délai de livraison des données. Ce phénomène de churn est donc important à étudier car cela influe sur la qualité de la vidéo reçue.

Nous étudions donc dans la figure B.15 la durée de vie des pairs qui ont communiqué avec nos noeuds.

La durée de vie des pairs varie largement pour toutes les applications même si la plupart des sessions vidéos sont courtes, surtout si on les compare à la durée totale d'un match de football. Cette observation est complètement différente des systèmes P2P de



(a) PPLive (b) PPSTream

(c) SOPCast (d) TVAnts

FIG. B.15 – Durée de vie des pairs pour toutes les applications mesurées.

TAB. B.7 – Fonction complémentaire de distribution cumulative (CCDF) et durée de vie moyenne des pairs.

	Lognormal $0.5 * erfc[\frac{log(x)-\mu}{\sigma*\sqrt{2}}]$	Weibull $e^{-(x/\beta)^\alpha}$	Moyenne Durée de vie (s)
PPLive	$\mu = 4.1, \sigma = 2.0$	$\alpha = 0.6, \beta = 162$	393
PPStream	$\mu = 5.4, \sigma = 2.1$	$\alpha = 0.6, \beta = 589$	1,222
SOPCast	$\mu = 6.1, \sigma = 2.1$	$\alpha = 0.6, \beta = 1,167$	1,861
TVANTS	$\mu = 7.2, \sigma = 1.3$	$\alpha = 0.9, \beta = 2,406$	2,778

partage de fichiers où les sessions peuvent s'étendre sur des dizaines d'heures ou de jours pour achever un téléchargement [87].

Nous avons également modélisé la durée de vie des pairs avec des distributions Log-

normal et de Weibull. Ainsi, nous pouvons modéliser la durée de vie des pairs pour de futures simulations avec des paramètres réalistes.

Finalement, les durées de vie des pairs sont relativement courtes ce qui indique un va-et-vient incessant des pairs dans ces systèmes P2P-TV comparé aux applications de partage de fichiers. La modélisation de la durée de vie des pairs avec des distribution de Weibull ou Log-normal permettra de modéliser ce paramètre de façon précise pour de futures simulations ou modèles. Cela pourrait être utile pour définir des mécanismes réalistes basés sur la durée de vie des pairs afin d'améliorer les partenariats entre les pairs.

Notre jeu de donnée est disponible pour la communauté de recherche [77]. Tout le monde peut ainsi se servir de ces traces et en extraire les valeurs exactes de la durée de vie des pairs pour obtenir des paramètres plus réalistes.

B.4 Mesures d'applications P2P-TV de part et d'autre du monde

Afin d'étendre la vision locale de notre précédente expérience de mesures, nous avons effectué une seconde campagne de mesures à plus large échelle en multipliant les points de mesures situés de part et d'autre du monde, en l'occurrence en France et au Japon. Notre objectif principal était d'obtenir une vision plus globale des réseaux P2P mesurés. L'idée sous-jacente était qu'en augmentant et diversifiant les points de mesures, nous pourrions obtenir une vue globale des réseaux P2P afin d'en comprendre les caractéristiques générales.

Cette expérience de mesure nous permet d'étudier l'organisation du réseau P2P et les échanges de trafic entre pairs, la distance à laquelle nos pairs obtiennent le contenu, de discuter de la situation géographique des utilisateurs et de l'impact du contenu diffusé sur cette population d'utilisateurs.

B.4.1 Expérience de mesures large-échelle

B.4.1.1 Motivation

Dans le cadre des applications P2P, les contenus sont transportés entre utilisateurs de différents FAI. Les contenus transportés sont généralement volumineux, notamment la vidéo qui consomme beaucoup de bande passante à des débits non élastiques. Les goulots d'étranglements de l'Internet sont supposés être situés au niveau des liens entre FAI et pas au niveau des liens du coeur de réseau d'un FAI [126]. Ainsi, le transport du trafic P2P est un véritable dilemme pour les FAI puisqu'il peut provenir de (et être échangé vers) n'importe quel autre FAI, sans que celui-ci ne connaisse la quantité exacte de données qui lui est transmise. Cela pourrait entraîner des coûts importants pour le FAI pour transporter ce trafic, mais surtout une impossibilité à dimensionner son réseau et à prévoir la charge de son réseau.

La prise en compte de la localité entre les pairs pour s'échanger le contenu est alors une problématique importante pour les FAI. Elle l'est également pour les utilisateurs, pour qui de longues distances entraîneraient souvent de longs délais de traversé de bout en bout du réseau, ce qui – pour du trafic vidéo à fortes contraintes temporelles – risquerait d'altérer la qualité de la vidéo reçue. En plus de la localité, les applications P2P nécessitent une contribution réciproque entre les pairs qui s'échangent les données. Il convient donc pour nous d'étudier à la fois la localité des pairs dans ces réseaux P2P mais également la quantité de données que les pairs s'échangent pour évaluer ce type d'application. Pour la localité, il en va de l'intérêt des FAI mais aussi d'une certaine façon de leurs utilisateurs. Pour la réciprocité des échanges, il en va de l'intérêt des utilisateurs et de la viabilité des solutions P2P à transmettre des contenus à très large échelle.

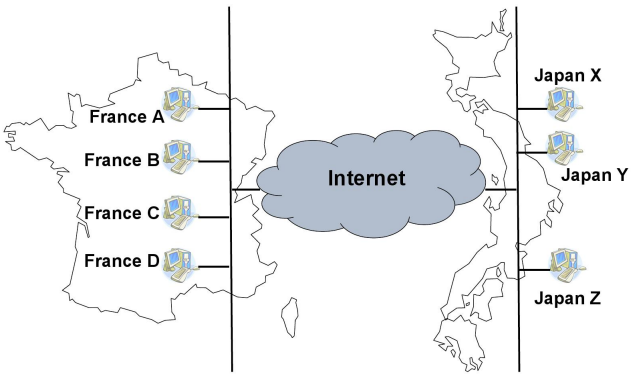


FIG. B.16 – Plate-forme expérimentale de mesures. Chaque PC est directement connecté à l'Internet à travers les réseaux du LIP6 et de WIDE.

B.4.1.2 Plate-forme expérimentale de mesures

Notre plate-forme expérimentale se décompose en deux parties situées de part et d'autre du monde : une partie en France et l'autre au Japon. Les parties Françaises et Japonaises sont identiques si ce n'est le nombre d'ordinateurs (PCs) que nous avions à notre disposition pour effectuer les mesures. Dans chacune des parties de la plate-forme, nous utilisions des PCs équipés de processeur cadencé à 1.8Ghz et de carte graphique standard. Tous les PCs étaient équipés de carte Ethernet à 100Mb/s. Les machines étaient directement connectées à l'Internet avec un accès Ethernet à 100Mb/s, à travers le réseau de notre université pour les machines situées en France, et le réseau du projet WIDE pour les machines situées au Japon. Le nombre de points de mesures (PCs) variait de trois à quatre en France, et deux à trois au Japon en fonction de la disponibilité de nos machines au moment de nos expériences. Tous les PCs utilisaient Windows XP comme système d'exploitation puisque les applications P2P-TV ont été principalement implémentées pour celui-ci. Les applications utilisent des codecs vidéo de type MPEG4. Chaque PC (noeud) utilisait Windump [78] pour collecter le trafic.

La plate-forme expérimentale de mesure est représentée sur la figure B.16. Dans un soucis de clarté, nous ferons référence aux PCs situés en France en les nommant *France A, France B France C et France D* et ceux situés au Japon *Japan X, Japan Y et Japan Z.*

B.4.1.3 Déroulement des expériences

Au cours de ces expériences, nous avons utilisé les applications les plus populaires, en se référant aux impressions de leurs utilisateurs via différents sites web ou forums [75], ainsi qu'à nos propres impressions générales sur ces applications et le niveau de qualité

que nous expérimentions avec celles-ci. Ainsi, nous avons choisi de mesurer le trafic de PPSTream, TVUPlayer, SOPCast et TVAnts [15–17, 19].

Pour chaque expérience, nous avons utilisé la même application P2P-TV sur toutes les machines de la plate-forme de mesure. Au cours de nos expériences, nous nous sommes intéressées à des évènements en direct et plus particulièrement à des évènements sportifs car ils présentent un véritable intérêt à être regardé en direct. Nous avons effectué notre campagne de mesure au cours de l'été et l'automne 2007 car nous avions accès au réseau WIDE situé à l'université de Tokyo au Japon. Nous avons mesuré des évènements sportifs avec différentes applications parmi lesquels des matchs de football (FIFA) mais également les championnats du monde d'athlétisme (IAAF) ou la coupe du monde de rugby (IRB). À la fin de notre campagne de mesure, nous avions mesuré une cinquantaine d'évènements sportifs. Chaque évènement ayant été mesuré avec cinq à sept points de mesures (PCs), nous bénéficions d'une collection de plusieurs centaines de traces de trafic P2P-TV. La taille totale de notre collection de données dépasse les 1.5 TB.

Parmi tous les évènements mesurés, nous avons choisi de nous concentrer sur des matchs de football puisqu'il existe une importante communauté d'utilisateurs qui utilise ces applications dans ce but précis. De notre collection de traces, nous avons extrait quatre expériences de mesure représentatives de notre jeu de données. Chacune d'entre elles a été effectuée avec l'une des quatre applications choisies.

Nous avons mesuré PPSTream avec cinq ordinateurs (noeuds) : trois était situés en France et deux au Japon. Le trafic a été collecté le 15 août 2007 lors du tour préliminaire de la ligue des champions de l'UEFA entre Liverpool et Toulouse. Le trafic de TVUPlayer a été collecté avec six noeuds répartis équitablement en France et au Japon. L'évènement mesuré était un match de qualification pour le tournoi olympique de football entre le Japon et le Vietnam le 22 août 2007. Pour SOPCast, six noeuds étaient à nouveau disponible (quatre en France et deux au Japon) et l'évènement était un autre match de qualification pour le tournoi olympique de football entre la Chine et le Japon. Ce match s'est déroulé le 3 août 2007. Enfin, TVAnts a pu être mesuré par l'ensemble de nos sept noeuds (quatre en France et trois au Japon). Il s'agissait encore d'un match qualificatif pour le tournoi olympique de football entre la Chine et le Vietnam, le 23 août 2007.

Le tableau 4.1 présente les traces étudiées. Toutes les traces ont la même durée de 2h45 (165 mn), plus longue qu'un match de football afin de capturer les éventuels effets des débuts et fins de transmission comme les arrivées massives d'utilisateurs en début de rencontre. Ces traces sont publiquement disponibles sur notre serveur d'archivage de données [77].

B.4.2 Observations générales

Dans cette section nous présentons les traces que nous allons étudier. Puisqu'il s'agit de 24 traces provenant de quatre applications distinctes et de nombreux points de mesures, nous allons adopter la notation suivante pour faire référence à chaque trace en les nommant XXX_Yz où : (i) XXX sont les trois premières lettres du nom

de l'application (PPS pour PPSTream, TVU pour TVUPlayer, SOP pour SOPCast et TVA pour TVAnts) ; (ii) *Y* est la première lettre du pays dont la trace provient (*F* pour la France et *J* pour le Japon) ; (iii) *z* est l'identifiant des noeuds dans chaque pays (*a, b, c, d* pour les noeuds de France et *x, y, z* pour ceux du Japon). Par exemple, la trace collectée par le noeud France C pour PPSTream est appelée PPS_Fc ; celle de Japan Y pour SOPCast est SOP_Jy.

Le tableau 4.1 présente la taille de chaque trace, leur nombre de paquets et d'adresses IPs qu'elles contiennent, ainsi que la proportion de trafic montant (*upload*) et descendant (*download*) en fonction du protocole de transport TCP ou UDP utilisé. Nous avons également ajouté les "IP vidéos", qui sont les IPs qui ont réellement échangé du trafic vidéo avec nos noeuds. Nous sommes parvenus à identifier ces IPs en utilisant l'heuristique proposée par Hei [84], brièvement rappelée ici : si une IP a échangé au moins 10 paquets d'au moins 1000 octets, alors on considère que cette IP a bien échangé du trafic vidéo. Les adresses IPs présentes dans nos traces sont en fait les pairs avec lesquels nous avons échangé du trafic et nous parlerons d'adresses IPs ou de pairs indifféremment par la suite.

Pour chaque trace, nous avons aussi calculé la proportion d'adresses IP présente dans les autres traces de la même application (*IP similarité*) afin d'évaluer l'effet de la multiplication de nos points de mesures sur les mesures elles-mêmes (tableau 4.2. Par exemple, PPS_Fb compte 3 317 IPs dont 57% sont également présentes dans les autres traces de PPSTream. Ainsi nous souhaitions vérifier si le nombre de points de mesures utilisés (5 à 7 dans notre cas) suffisait pour obtenir une vision globale du réseau P2P ou si au contraire il aurait fallu en utiliser un nombre plus important. En moyenne, pour PPSTream – mesuré avec 5 noeuds – 61% des IPs se retrouvent dans les autres traces. La similarité IP moyenne est de 75% pour TVUPlayer et 79% pour SOPCast (6 points de mesures) et 97% pour TVAnts (7 points de mesures). Ainsi, nous observons que plus nous augmentions le nombre de points de mesures, plus nos noeuds communiquaient avec l'ensemble des pairs du réseau, nous permettant d'obtenir une vue globale et précise du réseau P2P. Compte tenu du fait que 7 points de mesures permettent d'obtenir 97% d'IP similaires, nous pensons que cet ordre de grandeur du nombre de sondes est suffisant pour mesurer le réseau P2P dans sa globalité. Il semble donc inutile de le mesurer via un très grand nombre de pairs (centaines, milliers) ou même la totalité, expérience de toute façon quasi-impossible à réaliser pour un réseau P2P de cette échelle. Les expériences réalisées avec 5 ou 6 noeuds ne rendent pas exactement compte de la vision complète du réseau P2P mais permettent quand même de mieux caractériser le réseau P2P dans son ensemble, comparé aux précédents travaux de mesures effectués avec une unique sonde de mesure.

Les traces de PPSTream présentent toutes les mêmes caractéristiques à l'exception du nombre d'IPs qui diffère légèrement pour l'une des traces Françaises et Japonaises (PPS_Fa et PPS_Jy). Les traces Japonaises de TVUPlayer et SOPCast sont plus volumineuses que les Françaises. Ces applications comptent toutefois chacune le même nombre d'IPs dans chaque trace à l'exception de TVU_Fa. À l'inverse, les traces Françaises de TVAnts sont légèrement plus volumineuses que les traces Japonaises et comptent le même nombre d'IPs.

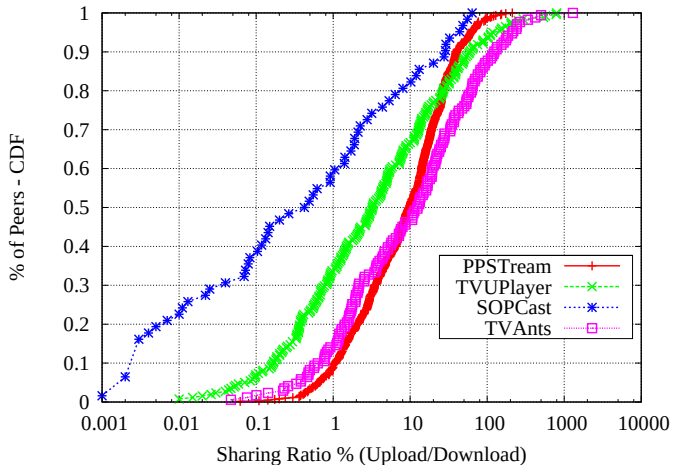
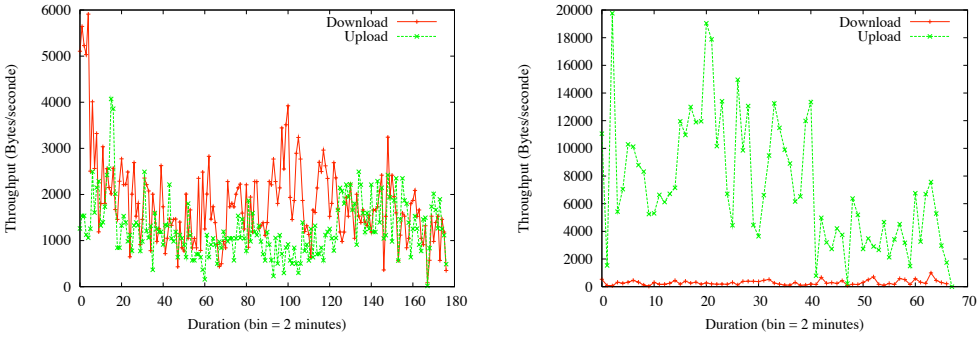


FIG. B.17 – Proportion de trafic partagé pour tous les pairs (Upload/Download).

Il est intéressant de constater que les traces de TVUPlayer ou de SOPCast, bien que plus petites que celles de PPSTream ou TVAnts, contiennent pourtant beaucoup plus de paquets. En approfondissant cette question, nous avons pu constater que 80% des paquets de TVUPlayer, par exemple, ont une taille inférieure à 200 octets alors que près de 2/3 des paquets de PPSTream ont une taille supérieure à 1000 octets. Cela est certainement dû à l'utilisation du protocole de transport UDP.

Chaque pair a comme objectif de télécharger la vidéo une seule fois mais peut la retransmettre plusieurs fois. Les traces volumineuses témoignent ainsi d'une importante activité en upload comme TVU_Jy, TVU_Jz ou SOP_Jx et SOP_Jy. Cependant si l'on compare les quantités de données téléchargées, elles varient pour TVUPlayer de 512MB pour TVU_Fb à 682MB pour TVU_Jz. C'est une différence assez importante et nous rappelons que toutes les traces ont la même durée de collecte de trafic (2h45). Cette quantité de trafic supplémentaire provient de la quantité de trafic de signalisation reçu. Un pair qui transmet beaucoup de vidéo reçoit également beaucoup de trafic de signalisation des autres pairs du réseau qui coordonnent leur réception. Cette observation somme toute anodine au premier abord peut en fait s'avérer être un véritable problème dans certaines situations. Un pair altruiste qui transmet beaucoup de vidéo à d'autres pairs se verrait alors gratifier d'une quantité importante de trafic de signalisation. Cela aurait pour effet de gaspiller sa bande passante pour télécharger la vidéo et saturer son lien d'accès, affectant probablement la qualité de la vidéo reçue. C'est d'autant plus problématique pour une application comme TVUPlayer qui utilise UDP, lui-même ne proposant aucun mécanisme de contrôle de congestion pour adapter les débits des flots vidéos à la capacité du lien.



(a) Échange équitable (b) Échange non équitable

FIG. B.18 – Échange de trafic entre différents pairs.

B.4.3 Échange de trafic entre les pairs

Dans cette section, nous nous intéressons à la quantité de trafic échangé par les pairs que nous contrôlons avec chacun des autres pairs de l'Internet. Au-delà du trafic échangé vers chaque pair dans chaque direction, il est important de mettre en relation la quantité de trafic envoyé et reçu pour chacun de ces pairs. Nous appelons cette quantité "la proportion de trafic partagé" (*sharing ratio*), qui est la quantité de trafic envoyé vers un seul pair (upload) divisé par la quantité de trafic reçu par ce même pair (download).

La figure B.17 présente la fonction de distribution cumulative (CDF) du trafic partagé entre les pairs que nous contrôlons et tous les autres pairs. Nous n'avons choisi qu'une trace pour chaque application car nous avons pu constater qu'elles étaient chacune représentatives de toutes les autres. Quand la proportion de trafic partagé est supérieure à 1, nos pairs sont altruistes avec les autres pairs et transmettent beaucoup de données, quand cette proportion vaut 1 alors l'échange de trafic a été équitable. Quand la proportion est inférieure à 1, alors nos pairs ont d'avantage reçu des autres pairs distants.

Pour PPSTream et TVAnts, seulement 10% des pairs sont altruistes avec nos pairs et ont transmis plus de données qu'ils en ont reçues. Les 90% de pairs restants ont bénéficié de notre altruisme. Ce comportement est similaire pour TVUPlayer à des proportions légèrement différentes : 30% des pairs de l'Internet ont beaucoup transmis à nos pairs et les 70% restant ont bénéficié des ressources de nos pairs. C'est le contraire pour SOPCast : la majorité des pairs de l'Internet ont beaucoup plus transmis aux pairs que nous contrôlons (60%). Dans tous les cas, les échanges de trafic pour toutes les applications ne sont pas équitables avec nos pairs. Une large majorité des pairs rencontrés sont principalement consommateurs de données (proportion de partage >1) plutôt que producteurs (< 1). Il s'agit également d'échanges non équitables pour SOPCast mais dans le sens inverse.

Toutefois, il existe différents comportements de pairs. Sur les figures B.18(a) et B.18(b)

nous avons isolé pour PPSTream le trafic d'upload et de download entre l'un de nos pairs et deux pairs distincts rencontrés sur l'Internet. Dans la figure B.18(a), les deux courbes de trafic ne sont pas identiques mais indiquent bien que les deux pairs ont essayé de contribuer dans les mêmes proportions au cours de leur échange. La figure B.18(b) contraste bien avec ce premier cas : on peut y voir que notre pair contribue énormément alors qu'il ne reçoit pratiquement rien en retour. D'après la proportion de trafic partagé (figure B.17), clairement, quelles que soient les applications où le noeud que nous observons, ce deuxième cas est celui qui se produit le plus fréquemment : les échanges entre pairs ne sont pas équitables et sans aucune réciprocité.

Cela indique qu'il n'y a pas d'équité entre pairs dans ces systèmes P2P-TV. Puisque les pairs sont responsables de retransmettre les données à d'autres pairs, chaque pair devrait transmettre au moins la même quantité de données qu'il reçoit de façon à produire autant qu'il consomme. Comme ce n'est pas le cas, cela pose le problème important du passage à l'échelle de ce type de systèmes. Puisqu'un système P2P d'échange de fichiers comme BitTorrent doit principalement son succès à l'efficacité de son mécanisme d'incitation qui encourage les pairs à contribuer réellement entre eux, il est alors urgent de concevoir ce type de mécanisme adapté aux systèmes P2P de diffusion vidéo.

B.4.4 Distance entre les pairs

Dans cette section, nous nous intéressons à la distance à laquelle nos pairs échangent du trafic avec les autres pairs de l'Internet. Pour cela, nous avons analysé le champ "Time-to-Live" (TTL) des datagrammes IP, qui est décrémenté par les routeurs à chaque saut dans l'Internet. Comme nous l'avons précisé, les applications utilisées sont toutes implémentées principalement pour Windows XP et requièrent ce système d'exploitation. La valeur du TTL par défaut de Windows XP est de 128. Ainsi, en comparant la valeur des TTL reçus à la valeur par défaut fixée par Windows XP, nous pouvons déduire le nombre de sauts auquel nos pairs reçoivent le trafic. Par exemple, si un datagramme IP nous arrive avec un TTL à 120, la valeur par défaut étant de 128, nous pouvons estimer la distance de l'émetteur du datagramme à 8 sauts. Cette valeur n'est qu'une estimation puisque le datagramme peut très bien avoir traversé une série d'équipements qui ne décrémente pas le champ TTL des datagrammes IP. Toutefois, cette heuristique est déjà communément utilisée [112] et nous permet d'estimer la distance des pairs avec lesquels nous communiquons. Il est à noter que l'on ne peut utiliser cette heuristique pour le trafic montant puisque les datagrammes que nous envoyons ont tous le même TTL par défaut et nous ne pouvons pas connaître la valeur de ce TTL lorsque le datagramme envoyé atteint sa destination.

Sur la figure B.19, nous avons tracé la fonction de distribution cumulative (CDF) de la distance en nombre de sauts. Nous n'avons représenté qu'une seule des applications, TVAnts, car les résultats étaient similaires pour chacune des autres applications.

On remarque très nettement que la distance de téléchargement des pairs Français est plus grande que celle des pairs Japonais. Pour nos pairs situés en France, 50% des pairs chez qui ils téléchargent la vidéo sont situés à plus de 20 sauts et 90% sont situés

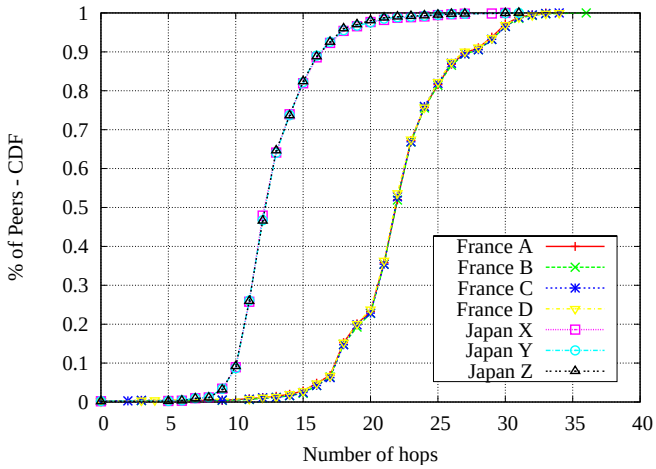


FIG. B.19 – Distance en nombre de sauts des pairs sur l'Internet (CDF).

à moins de 27 sauts. Pour nos noeuds Japonais, 50% des pairs distants sont situés à plus de 12 sauts seulement et 90% sont à moins de 16 sauts.

Plusieurs raisons peuvent expliquer ce phénomène. Tout d'abord, les applications P2P-TV ont été déployées par des sociétés Chinoises et sont déjà très populaires en Asie. La plupart des serveurs sources de vidéo ainsi que les utilisateurs sont probablement situés sur ce continent. L'utilisation d'applications P2P-TV en Europe est encore limitée et même PPSTream a le même comportement que les autres applications alors qu'un match de football présentant un fort intérêt pour les utilisateurs Européens était diffusé (Ligue des champions UEFA). Par conséquent, si la très grande majorité des utilisateurs est située en Asie, les noeuds Français doivent récupérer le contenu à partir de pairs situés à une distance plus lointaine comparé aux noeuds Japonais. Ensuite, les applications P2P-TV n'utilisent apparemment pas de mécanisme de prise en compte de la localité pour sélectionner les pairs qui fourniront le contenu. Si cela était le cas, un scénario possible serait qu'un noeud Français télécharge la vidéo d'une distance lointaine et la retransmet aux autres noeuds Français situés dans le même réseau. Dans ce cas, on aurait observé sur la figure B.19 un seul pair Français qui télécharge la vidéo à une distance lointaine et les autres pairs auraient reçu les données d'une plus proche distance.

Le téléchargement de la vidéo a une lointaine distance est un problème important puisque cela aura pour effet d'augmenter le délai de bout en bout pour recevoir les paquets vidéos, ce qui affectera la qualité de la vidéo. Cela révèle également le fait que le trafic transite entre différents FAI et traverse les liens entre FAI pour atteindre les destinations lointaines.

B.4.4.1 Systèmes autonomes

Il nous a ensuite semblé nécessaire d'étudier le trafic à une granularité supérieure c'est-à-dire à l'échelle des systèmes autonomes (AS). Un système autonome est un ensemble de réseau IP sous le contrôle d'une seule et même entité, typiquement un FAI. Le réseau d'un FAI peut quant à lui être subdivisé en plusieurs AS. La quantité d'AS traversés peut donner des informations sur la façon dont le trafic est échangé entre différents FAI. Clairement, plus il y a d'AS impliqués dans les échanges ou plus le nombre d'AS traversés consécutivement est important (nombre de sauts d'AS), plus cela témoigne d'échanges de trafic entre différents FAI.

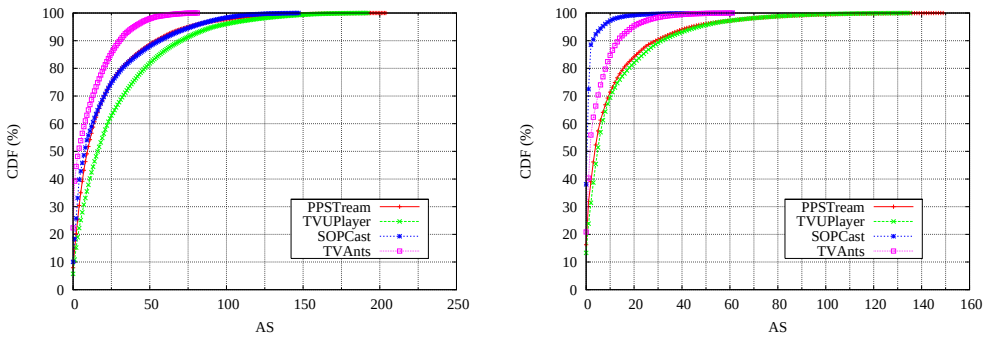
Ainsi, en utilisant le service `whois` via les serveurs de cymru [116], nous avons obtenu la correspondance entre les adresses IPs et leur AS. À nouveau, nous ne nous intéressons qu'aux IPs vidéos car ce sont les seules à réellement échanger du trafic.

La figure B.20 présente la CDF du volume de trafic échangé avec chaque AS ; chacune des sous-figures est pour l'upload et le download. Nous ne présentons à nouveau qu'une trace par application car elles sont représentatives des autres traces. Les AS sont ordonnés par ordre croissant du volume échangé sur l'axe des abscisses : l'AS 0 est celui qui a échangé le plus de trafic avec l'un de nos pairs. Afin de lever toute ambiguïté, nous insistons sur le fait que l'axe des abscisses représente bien le classement des AS dans l'ordre du trafic échangé et n'est en aucune manière relative aux identifiants des AS sur l'Internet.

On dénombre beaucoup plus d'AS en upload qu'en download. Toutefois, quelle que soit la direction du trafic, des centaines d'AS sont impliqués dans les échanges. PPSTream compte par exemple 200 AS en upload, ce qui est le double de TVAnts. Seuls 60 AS suffisent à PPSTream et SOPCast pour échanger 90% du trafic montant ; toutefois SOPCast échange du trafic vers moins d'AS que PPSTream (150 AS). TVUPlayer utilise autant d'AS en upload que PPSTream mais son trafic semble mieux réparti dans chaque AS : 90% du trafic montant de TVUPlayer est dirigé vers 75 AS différents. TVAnts utilise moins d'AS comparé aux autres : 90% de son trafic montant est dirigé vers seulement 30 AS, soit moitié moins que les autres applications. Du point de vue du téléchargement, le trafic provient d'un plus faible nombre d'AS quelle que soit l'application. SOPCast télécharge la majorité du trafic de seulement trois AS, TVAnts en utilise une dizaine alors que TVUPlayer et PPSTream en utilise une trentaine.

Afin d'obtenir le nombre de sauts en termes d'AS entre les pairs que nous contrôlons et les autres pairs de l'Internet, nous avons procédé comme il suit : pour certaines de nos expériences de mesures, nous avions effectué un `traceroute` vers chaque destination. Ensuite, pour chaque interface réseau traversée, nous avons – comme précédemment – obtenu l'AS d'origine d'une adresse IP en utilisant le service de correspondance entre une adresse IP et son AS fourni par cymru (`whois`). En regroupant par AS, il nous était alors possible d'obtenir le nombre d'AS traversés entre nos pairs et la destination. Pour ces expériences, nous avons effectué les `traceroute` à partir de nos pairs Français uniquement. Ils sont en effet l'objet de notre attention car ils téléchargent la vidéo à une distance lointaine en termes de sauts comparé aux pairs Japonais.

Sur la figure B.21, nous présentons la CDF du trafic échangé en fonction du nombre



(a) Upload (b) Download

FIG. B.20 – Volume de trafic (CDF) échangé vers chaque système autonome (AS).

de sauts d'AS. Dans un soucis d'économie de place, nous présentons les résultats pour SOPCast, mais les résultats sont similaires pour toutes les autres traces et applications. L'axe des abscisses représente le nombre de sauts en termes d'AS et l'axe des ordonnés le cumul du trafic échangé.

On remarque clairement que le nombre de sauts d'AS en download est plus important que le nombre de sauts en upload. Par exemple, 50% du trafic est envoyé (upload) à 3 sauts d'AS ou moins alors que l'on ne télécharge (download) que 30% du trafic à cette distance. 15% du trafic est également téléchargé d'une distance comprise entre 5 et 7 sauts d'AS alors que le trafic envoyé n'atteint jamais de destinations à plus de 5 sauts d'AS. Cela indique à nouveau que nos pairs Français téléchargeaient le trafic loin d'eux ; par contre ils le réémettaient plus proche d'eux.

Précédemment, nous avons observé que les noeuds situés en France téléchargent la vidéo à une distance plus lointaine que les noeuds Japonais (figure B.19). Le trafic échangé entre pairs implique un nombre important d'AS (figure B.20) et, pour les noeuds Français, le trafic téléchargé a transité à travers plus d'AS comparé au trafic envoyé (figure B.21). Ces résultats illustrent le caractère problématique du trafic P2P-TV pour les FAI. Les systèmes P2P-TV ne tiennent pas compte de la localité des pairs pour échanger le trafic et cela engendre de considérables difficultés d'ingénierie de trafic pour les FAI. Du point de vue des FAI, il est important de maintenir le trafic P2P local à leur propre réseau, sinon cela risquerait de surcharger les liens entre FAI, qui sont déjà considérés comme étant les goulots d'étranglement de l'Internet [126]. Les liens entre FAI sont également très coûteux et les FAI souhaitent limiter leur utilisation à leurs propres clients.

B.4.4.2 Localité géographique

Dans cette section, nous étudions la localité géographique des pairs du réseau. Nous avons effectué la correspondance entre adresses IPs et pays d'origine en interrogeant la base de données GeoIP MaxMind [118]. Nous présentons alors ces résultats sous forme

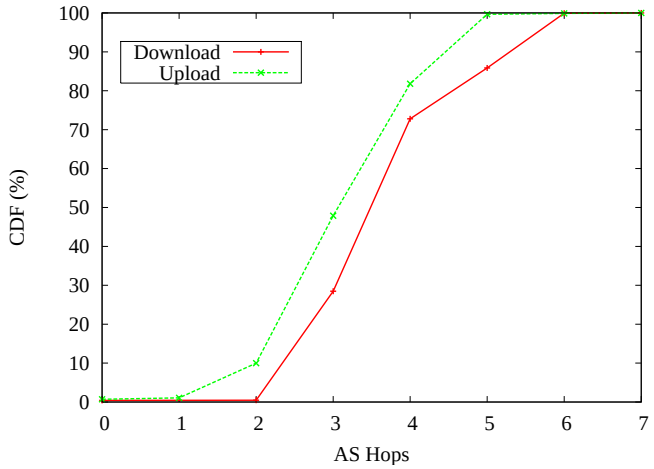
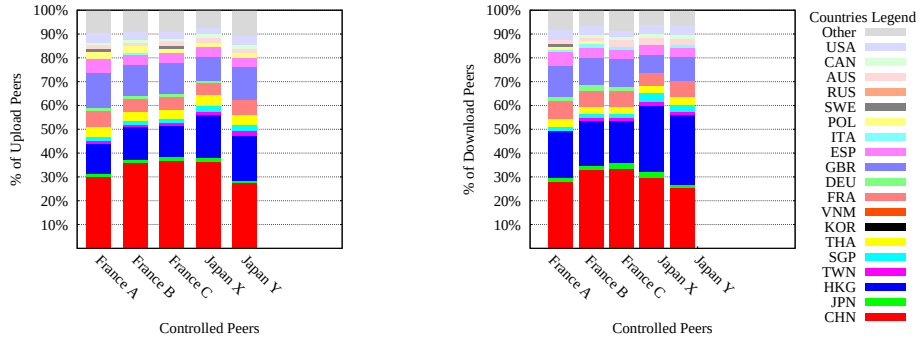


FIG. B.21 – Volume de trafic (CDF) en fonction du nombre de sauts d'AS (systèmes autonomes).

d'histogrammes empilés sur la figure B.22. Chaque application est représentée par deux figures, l'une pour la proportion de pairs en upload, l'autre pour le download. Sur chaque figure, nous présentons autant d'histogrammes qu'il y avait de traces par application. Nous présentons dans la légende des histogrammes les pays qui ont représenté une part importante du trafic dans toutes nos traces. Les autres pays contribuaient pour moins de 1% dans un faible nombre de traces et sont cumulés dans la catégorie "autre" (*other*).

Comme il était attendu, le contenu diffusé a un impact sur la population de ces systèmes. Par exemple, on observe un nombre important de pairs de Grande-Bretagne ou de France avec PPSTream. Il y a également un nombre important de pairs de Chine avec SOPCast et TVAnts ou du Japon avec SOPCast et TVUPlayer, c'est-à-dire lorsque le contenu diffusé impliquait ces pays. La proportion de pairs Chinois et Japonais est moins importante avec les deux autres applications. De plus, nous avons pu observer la présence de pairs localisés au Vietnam avec TVUPlayer. Même si nous n'en avons pas observé pour TVAnts, il était somme toute surprenant d'en apercevoir compte tenu des conditions présumées mauvaises d'accès à l'Internet de ce pays encore émergeant économiquement [127]. On peut aussi remarquer que chaque application compte – en download – une grande partie de leurs pairs situés en Asie. En upload, par contre, nos pairs situés en France semblent compter d'avantage de pairs en Europe que les pairs Japonais, qui eux en compte plus en Asie. Cela confirme nos précédentes observations de nos pairs Français qui téléchargent le trafic plus loin que les pairs Japonais mais le réémettent plus proche d'eux.



(a) PPSTream Upload (b) PPSTream Download

(c) TVUPlayer Upload (d) TVUPlayer Download

(e) SOPCast Upload (f) SOPCast Download

(g) TVAnts Upload (h) TVAnts Download

FIG. B.22 – Localisation géographique des pairs pour PPStream (a,b), TVUPlayer (c,d), SOPCast (e,f) et TVAnts (g,h).

B.5 Conclusion et Perspectives

En partant du web et ses documents hypertext au début des années 1990 aux nouveaux services populaires tels que le partage de fichiers, la voix sur IP, les jeux en ligne ou les services vidéos, l'Internet est devenu le principale médium de livraison de tout type de contenu aux utilisateurs. En particuliers, ces dernières années, d'importants efforts ont été menés pour concevoir de nouvelles architectures pour la diffusion de flots multimédias en direct à large échelle, afin d'offrir un service de télévision aux utilisateurs de l'Internet. Ces nouvelles architectures utilisent le principe du P2P, qui a déjà montré son efficacité à distribuer du contenu à large échelle.

Dans cette thèse, nous avons tout d'abord présenté les principales architectures P2P diffusion de vidéo en direct. La plupart de ces architectures n'ont pas été implémentées, expérimentées ou déployées sur l'Internet. En parallèle, de nombreuses applications P2P-TV commerciales ont été déployées sur l'Internet. Ces applications sont propriétaires et ne fournissent aucun détail d'implémentation.

Au début de ces travaux de thèse, les systèmes P2P-TV étaient méconnus et aucune information n'était disponible pour décrire ces systèmes. Ainsi, afin de combler l'écart entre les architectures académiques et les applications commerciales, nous avons effectué d'importantes expériences de mesures de systèmes P2P-TV populaires. En mesurant le trafic réseau d'applications P2P-TV populaires, nous avons obtenu une profonde connaissance de ces systèmes, mécanismes et propriétés. Nous avons également étudié l'impact de ce trafic sur les réseaux.

Pour résumer, nous avons mis en évidences les similarités et différences parmi les principales applications P2P-TV, ainsi que déduit leurs mécanismes et autres propriétés. Nous pensons que les résultats de nos travaux permettront l'utilisation de données réalistes et fiables afin de simuler ou modéliser ces systèmes dont la popularité ne fait que croître.

D'ailleurs, comme ces applications sont en pleines expansions, il était également judicieux de les surveiller en mesurant leur trafic.

À la suite de notre première expérience de mesure, d'autres travaux de mesures sont apparus. Nous avons présenté les principaux en les classant suivant la technique utilisé pour mesurer le trafic. La plupart de ces expériences n'avaient qu'une vision locale des systèmes P2P-TV mesurés et se concentrait uniquement sur leurs mécanismes. Ainsi, pour étendre nos précédents travaux de mesure, nous avons effectué une seconde campagne de mesure à plus large échelle en mesurant le trafic d'application P2P-TV de part et d'autre du monde, c'est à dire en France et au Japon. Grâce à cette nouvelle expérience, nous étions en mesure d'obtenir une vision globale du réseau P2P et de mieux caractériser le réseau P2P lui-même ainsi que l'organisation des pairs.

Puisque le point de départ de nos travaux était de comprendre les systèmes P2P-TV, nous pensons que nous avons atteint notre but grâce à ces différentes expériences de mesure.

Comme perspective majeurs de nos travaux, en se basant sur nos mesures d'applications P2P-TV, nous songeons à définir de nouveaux mécanismes adaptés à ce type de diffusion. Ces mécanismes peuvent être des mécanismes d'incitation afin d'encoura-

ger tous les pairs à réellement collaborer dans le réseau. Cela peut aussi concerner des mécanismes de prises en compte de la localité afin d'améliorer la qualité de la vidéo transmise et permettre aux fournisseurs d'accès à Internet de mieux prédire la charge de trafic sur leur réseau.

Bibliography

[1] M. Knights, "Web 2.0," *Communications Engineer*, vol. 5, no. 1, pp. 30–35, February-March 2007.

[2] I. V. Yakovlev, "Web 2.0: Is it evolutionary or revolutionary?" *IT Professional*, vol. 9, no. 6, pp. 43–45, Nov.-Dec. 2007.

[3] S. Murugesan, "Understanding web 2.0," *IT Professional*, vol. 9, no. 4, pp. 34–41, July-Aug. 2007.

[4] R. Jain, "I want my iptv," *Multimedia, IEEE*, vol. 12, no. 3, pp. 96–96, July-Sept. 2005.

[5] D. H. Shin, "Potential user factors driving adoption of iptv. what are customers expecting from iptv?" *Technological Forecasting and Social Change*, vol. 74, no. 8, pp. 1446 – 1464, 2007.

[6] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network - measurements, models, and implications," *Elsevier Computer Networks*, 2009.

[7] S. Boll, "Multitube–where web 2.0 and multimedia could meet," *Multimedia, IEEE*, vol. 14, no. 1, pp. 9–13, Jan.-March 2007.

[8] B. Garfield, "The youtube effect," *Wired Magazine*, December 2006. [Online]. Available: http://www.wired.com/wired/archive/14.12/youtube.html

[9] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2007, pp. 1–14.

[10] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, and S. Tewari, "Will iptv ride the peer-to-peer stream? [peer-to-peer multimedia streaming]," *Communications Magazine, IEEE*, vol. 45, no. 6, pp. 86–92, June 2007.

[11] "Akamai." [Online]. Available: http://www.akamai.com

[12] X. Zhang, J. Liu, B. Li, and T. P. Yum, "Coolstreaming/donet: A data-driven overlay network for peer-to-peer live media streaming," in *Proc. of IEEE Infocom*, 2005.

[13] M. Zhang, L. Zhao, Y. Tang, J.-G. Luo, , and S.-Q. Yang, "Large-scale live media streaming over peer-to-peer networks through global internet," in *ACM P2PMMS*, 2005.

[14] "Pplive." [Online]. Available: http://www.pplive.com

[15] "Ppstream." [Online]. Available: http://www.ppstream.com

[16] "Sopcast." [Online]. Available: http://www.sopcast.com

[17] "Tvants." [Online]. Available: http://www.tvants.com

[18] "Uusee." [Online]. Available: http://www.uusee.com

[19] "Tvuplayer." [Online]. Available: http://www.tvunetworks.com

[20] "Ppmate." [Online]. Available: http://www.ppmate.com

[21] B. Cohen, "Incentives build robustness in bittorrent," in *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[22] "edonkey," http://www.edonkey2000.com.

[23] "Youtube," http://www.youtube.com.

[24] http://www.dailymotion.com.

[25] "Internet studies." [Online]. Available: http://www.ipoque.com/ipoque-internet-study-08-09.pdf

[26] M. Hofmann and L. R. Beaumont, *Content Networking: Architecture, Protocols, and Practice*. Elsevier, 2005.

[27] [Online]. Available: http://www.multichannel.com/article/191223-YouTube_May_Lose_470_Million_In_2009_Analysts.php

[28] S. E. Deering, "Multicast routing in a datagram internetwork," Ph.D. dissertation, Stanford University, 1991.

[29] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the ip multicast service and architecture," *Network, IEEE*, vol. 14, no. 1, pp. 78–88, Jan/Feb 2000.

[30] "Napster." [Online]. Available: http://www.napster.com

[31] "Skype." [Online]. Available: http://www.skype.com

[32] M. Bawa, H. Deshpande, and H. Garcia-Molina, "Transience of peers & streaming media," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 107–112, 2003.

[33] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming live media over a peer-to-peer network," August 2002.

[34] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, vol. 32, no. 4. New York, NY, USA: ACM Press, October 2002, pp. 205–217.

[35] D. A. Tran, K. A. Hua, and T. Do, "Zigzag: an efficient peer-to-peer scheme for media streaming," vol. 2, 2003, pp. 1283–1292 vol.2.

[36] Y.-H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, vol. 28, no. 1. New York, NY, USA: ACM Press, June 2000, pp. 1–12.

[37] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat, "Bullet: High bandwidth data dissemination using an overlay mesh," in *SOSP'03: Proceedings of the 19th ACM Symposium on Operating Systems Principles*, 2003.

[38] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *ACM SIGCOMM*, vol. 26,4. New York: ACM Press, Aug. 1996, pp. 117–130.

[39] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74–93, September 2001.

[40] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: high-bandwidth multicast in cooperative environments," in *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM Press, 2003, pp. 298–313.

[41] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in communications (JSAC)*, vol. 20, no. 8, pp. 1489–1499, 2002.

[42] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," *Lecture Notes in Computer Science*, vol. 2218, pp. 329–350, 2001.

[43] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *Network Protocols, 2003. Proceedings. 11th IEEE International Conference on*, 2003, pp. 16–27.

[44] R. Rejaie and A. Ortega, "Pals: peer-to-peer adaptive layered streaming," in *NOSSDAV '03: Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video.* New York, NY, USA: ACM Press, 2003, pp. 153–161.

[45] T. A. Djamal-Eddine Meddour, Mubasher Mushtaq, "Open issues in p2p multimedia streaming," in *IEEE Internation Conference on Communication*, 2006.

[46] Z. Liu, Y. Shen, K. W. Ross, S. S. Panwar, and Y. Wang, "Substream trading: Towards an open p2p live streaming system," in *IEEE International Conference on Network Protocols*, 2008.

[47] Y. Shen, Z. Liu, S. S. Panwar, K. W. Ross, and Y. Wang, "Streaming layered encoded video using peers," in *IEEE International Conference on Multimedia & Expo*, 2005.

[48] T. Silverston and O. Fourmaux, "Source vs data-driven approach for live p2p streaming," in *Internation Conference on Networking*, April 2006.

[49] N. Magharei, R. Rejaie, and Y. Guo, "Mesh or multiple-tree: A comparative study of live p2p streaming approaches," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1424–1432, May 2007.

[50] M. Zhang, C. Chen, Y. Xiong, Q. Zhang, and S. Yang, "Optimizing the throughput of data-driven based streaming in heterogeneous overlay network," in *ACM Multimedia Modeling*, 2007.

[51] F. Picconi and L. Massoulié, "Is there a future for mesh-based live video streaming?" in *IEEE International Conference on Peer-to-Peer Computing (P2P'08)*, 2008.

[52] X. Hei, Y. Liu, and K. Ross, "Inferring network-wide quality in p2p live streaming systems," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 9, pp. 1640–1654, December 2007.

[53] "Mininova." [Online]. Available: http://www.mininova.org

[54] "The pirate bay." [Online]. Available: http://www.thepiratebay.com

[55] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in bittorrent?" in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2007.

[56] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer, "Free Riding in BitTorrent is Cheap," in *5th Workshop on Hot Topics in Networks (HotNets), Irvine, California, USA*, November 2006.

[57] A. Ganesh, A. Kermarrec, and L. Massoulie, "Peer-to-peer membership management for gossip-based protocols," in *IEEE Trans. Computers*, 2003.

[58] M. zhang, Q. Zhang, L. Sun, and S. Yang, "Understanding the power of pull-based streaming protocol: Can we do better?" in *IEEE Journal on Selected Areas in Communications*, 2007.

[59] F. Pianese, D. Perino, J. Keller, and E. W. Biersack, "PULSE: an adaptive, incentive-based, unstructured P2P live streaming system," *IEEE Trans. Multimedia, Vol. 9 N6, November*, 2007.

[60] A. Legout, G. Urvoy-Keller, and P. Michiardi, "Rarest first and choke algorithms are enough," in *Proc. of ACM IMC*, 2006.

[61] M. Wang and B. Li, "R2: Random push with random network coding in live peer-to-peer streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1655–1666, 2007.

[62] R. Ahlswede, N. Cai, S. yen Robert Li, R. W. Yeung, S. Member, and S. Member, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, 2000.

[63] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE International Symposium on Information Theory*, 2003.

[64] M. Wang and B. Li, "Lava: A reality check of network coding in peer-to-peer live streaming," in *IEEE Infocom*, 2007, pp. 1082–1090.

[65] N. Magharei and R. Rejaie, "Prime: Peer-to-peer receiver-driven mesh-based streaming," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1415–1423, May 2007.

[66] V. Venkataraman, K. Yoshida, and P. Francis, "Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast," *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, pp. 2–11, Nov. 2006.

[67] V. Vishnumurthy and P. Francis, "On overlay construction and random node selection in heterogeneous unstructured p2p networks," in *IEEE Infoco*, 2006.

[68] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, 2005.

[69] X. Zhang, J. Liu, and B. Li, "On large-scale peer-to-peer live video distribution: Coolstreaming and its preliminary experimental results," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2005.

[70] B. Li, S. Xie, Y. Qu, G. Y. Keung, C. Lin, J. Liu, and X. Zhang, "Inside the new coolstreaming: Principles, measurements and performance implications," in *IEEE Infocom*, 2008.

[71] B. Li, S. Xie, G. Y. Keung, J. Liu, I. Stoica, H. Zhang, and X. Zhang, "An empirical study of the coolstreaming+ system," *IEEE Journal on Selected Areas in Communications*, 2007.

[72] "Mediametrie." [Online]. Available: http://www.mediametrie.fr

[73] "Softonic." [Online]. Available: http://www.softonic.com

[74] [Online]. Available: http://www.mobilemarketingmagazine.co.uk/mobile_tv/index.html

[75] [Online]. Available: http://www.myp2p.eu

[76] Y. Huang, T. Z. J. Fu, D.-M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale p2p-vod system," in *ACM SIGCOMM*, 2008.

[77] [Online]. Available: http://content.lip6.fr/traces/

[78] [Online]. Available: http://windump.polito.it

[79] [Online]. Available: http://www.grid.unina.it/software/Plab

[80] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent packet traffic," *Networking, IEEE/ACM Transactions on*, vol. 4, no. 2, pp. 209–223, Apr 1996.

[81] T. Silverston and O. Fourmaux, "P2p iptv measurement: a comparison study," 2006. [Online]. Available: http://www.arxiv.org/abs/cs.NI/0610133

[82] P. Srisuresh and K. Egevang. Ietf rfc 3022: Traditional ip network address translator (traditional nat).

[83] [Online]. Available: http://www.joost.com

[84] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "Insights into pplive: A measurement study of a large-scale p2p iptv system," in *Proc. of IPTV Workshop, International World Wide Web Conference*, 2006.

[85] D. Veitch and P. Abry, "A wavelet-based joint estimator of the parameters of long-range dependence," *Information Theory, IEEE Transactions on*, vol. 45, no. 3, pp. 878–897, Apr 1999.

[86] ——, "Matlab code for the wavelet based analysis of scaling processes." [Online]. Available: http://www.cubinlab.ee.mu.oz.au/darryl

[87] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "Measuring and analyzing the characteristics of napster and gnutella hosts," *Multimedia Systems*, 2002.

[88] http://www.weibull.com.

[89] http://www.easyfit.com.

[90] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *IEEE Infocom*, 2006.

[91] Z. Liu, C. Wu, B. Li, and S. Zhao, "Distilling superior peers in large-scale p2p streaming systems," in *IEEE Infocom*, 2009.

[92] C. Rolland, J. Ridoux, and B. Baynat, "Litgen, a lightweight traffic generator: application to p2p and mail wireless traffic," in *Proceedings of the Passive and Active Measurement Conference (PAM 2007)*, Louvain-La-Neuve, Belgium, April 2007.

[93] S. Ali, A. Mathur, and H. Zhang, "Measurement of commercial peer-to-peer live video streaming," in *Proc. of Workshop in Recent Advances in Peer-to-Peer Streaming*, 2006.

[94] V. Aggarwal, S. Bender, A. Feldmann, and A. Wichmann, "Methodology for estimating network distances of gnutella neighbors," in *GI*, 2004.

[95] http://www.planet-lab.org.

[96] C. Wu, B. Li, and S. Zhao, "Multi-channel live p2p streaming: Refocusing on servers," in *IEEE Infocom*, 2008.

[97] K. Shami, D. Magoni, H. Chang, W. Wang, and S. Jamin, "Impacts of peer characteristics on p2ptv networks scalability," in *IEEE Infocom*, 2009.

[98] http://www.zattoo.com.

[99] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale p2p iptv system," in *IEEE Transactions on Multimedia*, 2007.

[100] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt, "Measurement and modeling of a large-scale overlay for multimedia streaming," in *Proc. of QSHINE'07, International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 2007.

[101] L. Vu, I. Gupta, K. Nahrstedt, and J. Liang, "Understanding overlay characteristics of a large-scale peer-to-peer iptv system," University of Illinois at Urbana-Champaign, Tech. Rep., September 2008.

[102] P. Dhungel, X. Hei, K. W. Ross, and N. Saxena, "The pollution attack in p2p live video streaming: Measurement results and defenses," in *ACM Sigcomm P2P-TV Workshop*, 2007.

[103] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," in *Proc. of ACM SIGCOMM IMW*, 2002.

[104] S. Tang, Y. Lu, J. M. Hernandez, F. Kuipers, and P. V. Mieghem, "Topology dynamics in a p2ptv network," in *IFIP Networking*, 2009.

[105] B. Fallica, Y. Lu, F. Kuipers, R. Kooij, and P. V. Mieghem, "On the quality of experience of sopcast," in *IEEE Future Multimedia Networking (FMN)*, 2008.

[106] Y. Lu, B. Fallica, F. Kuipers, R. Kooij, and P. V. Mieghem, "Assessing the quality of experience of sopcast," *International Journal of Internet Protocol Technology*, 2009.

[107] E. Alessandria, M. Gallo, E. Leonardi, M. Mellia, and M. Meo, "P2p-tv systems under adverse network conditions: a measurement study," in *IEEE Infocom*, 2009.

[108] Y.Zhang, N.Zong, G. Camarillo, J.Seng, and R.Yang, "Problem statement of p2p streaming protocol (ppsp)," http://www.ietf.org/internet-drafts/draft-zhang-ppsp-problem-statement-03.txt, May 2009.

[109] Y. Zhang, "Ietf internet-draft: Protocol analysis and comparison of pplive and ppstream by internet measurement," http://tools.ietf.org/html/draft-zhang-ppsp-protocol-comparison-measurement-00, September 2009.

[110] K.Cho, K. Fukuda, H.Esaki, and A.Kato, "The impact and implications of the growth in residential user-to-user traffic," in *ACM SIGCOMM*, 2006.

[111] long IKAN and NTT, "The proving ground of ntt," http://www.nyquistcapital.com/2007/08/28/the-proving-ground-of-ntt/, August 2007.

[112] http://www.cpni.gov.uk/Docs/InternetProtocol.pdf.

[113] F. Gont and P. Srisuresh, "Ietf internet-draft: Security implications of network address translators (nats)," http://tools.ietf.org/html/draft-gont-behave-nat-security-01.

[114] J. Hawkinson and T. Bates, "Ietf rfc1930: Guidelines for creation, selection, and registration of an autonomous system (as)," http://tools.ietf.org/html/rfc1930, 1996.

[115] "Ietf rfc 3912: Whois protocol specification," http://tools.ietf.org/html/rfc3912.

[116] http://www.cymru.com.

[117] "The gnu netcat," http://netcat.sourceforge.net/.

[118] "Maxmind geoip technology," http://www.maxmind.com.

[119] "Cygwin," http://cygwin.com.

[120] "Paris traceroute," http://www.paris-traceroute.net/.

[121] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with paris traceroute," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2006.

[122] B. Augustin, R. Teixeira, and T. Friedman, "Measuring load-balanced paths in the internet," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2007.

[123] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: A practical approach to reducing cross-isp traffic in peer-to-peer systems," in *ACM SIGCOMM*, 2008.

[124] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4p: Provider portal for applications," in *ACM SIGCOMM*, 2008.

[125] S. L. Blond, A. Legout, and W. Dabbous, "Pushing bittorrent locality to the limit," INRIA, Tech. Rep. inria-00343822, 2009.

[126] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can isps and p2p systems cooperate for improved performance?" *ACM SIGCOMM Computer Communications Review (CCR)*, July 2007.

[127] "Communications in vietnam." [Online]. Available: https://www.cia.gov/library/publications/the-world-factbook/geos/VM.html

[128] http://peersim.sourceforge.net/.

[129] Y. Guo, C. Lin, H. Yin, and Z. Zhao, "Design and analysis of iptv digital copyright management security protocol," in *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on*, December 2007.

[130] Z. Hua, C. Chunxiao, Z. Li, Y. Shiqiang, and Z. Lizhu, "Content protection for iptv-current state of the art and challenges," in *Computational Engineering in Systems Applications, IMACS Multiconference on*, Oct. 2006.

[131] H. Singh, C. Kvvon, S. S. Kim, and C. Ngo, "Iptv over wirelesslan: Promises and challenges," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, Jan. 2008, pp. 626–631.

[132] F. Retnasothie, M. Ozdemir, T. Yucek, H. Celebi, J. Zhang, and R. Muththaiah, "Wireless iptv over wimax: Challenges and applications," in *Wireless and Microwave Technology Conference, 2006. WAMICON '06. IEEE Annual*, December 2006.

[133] J. Simsarian and M. Duelk, "Iptv bandwidth demands in metropolitan area networks," June 2007.

[134] G. Goth, "Redefining the server as home networks emerge," *Internet Computing, IEEE*, vol. 12, no. 2, pp. 7–9, March-April 2008.

[135] J. Kishigami, "The role of qoe on iptv services style," in *Multimedia, 2007. ISM 2007. Ninth IEEE International Symposium on*, Dec. 2007.

[136] M. Zapater and G. Bressan, "A proposed approach for quality of experience assurance of iptv," in *Digital Society, 2007. ICDS '07. First International Conference on the*, Jan. 2007.

[137] C.-S. Lee, "Iptv over next generation networks in itu-t," in *Broadband Convergence Networks, 2007. BcN '07. 2nd IEEE/IFIP International Workshop on*, May 2007.

List of Figures

2.1 Client-server . . . 6
2.2 CDN servers . . . 7
2.3 Multicast . . . 8
2.4 Peer-to-Peer . . . 8
2.5 Single-tree approach . . . 10
2.6 Peercast . . . 10
2.7 Impact of the departure of a peer. . . . 11
2.8 NICE . . . 12
2.9 Narada . . . 13
2.10 Bullet . . . 14
2.11 Second-generation protocols . . . 15
2.12 Splitstream . . . 16
2.13 Mesh-based architectures . . . 18
2.14 BitTorrent . . . 20
2.15 DONet architecture and topology . . . 21
2.16 Pulse . . . 23
2.17 R^2 . . . 24
2.18 PRIME . . . 25

3.1 Measurement experiments testbed . . . 34
3.2 Example of throughput for TVAnts . . . 36
3.3 Average packet size according to the session duration . . . 37
3.4 Joint probability distribution of Inter Packet Time and Packet Size . . . 39
3.5 Upstream flows: Average Packet Size vs. Number of Packets . . . 40
3.6 Packet rate of video and signaling traffic . . . 42
3.7 PPLive: Traffic energy spectra . . . 44
3.8 PPSTream: Traffic energy spectra . . . 45
3.9 SOPCast: Traffic energy spectra . . . 46
3.10 TVAnts: Traffic energy spectra . . . 47
3.11 Bytes sent vs. Bytes received for each host . . . 48
3.12 Video download policies . . . 50

3.13 Peers neighborhood for all the applications 51
3.14 Video peers lifetime 53

4.1 Large-scale measurement experiments testbed 66
4.2 Traffic by peers 74
4.3 Sharing ratio between peers 75
4.4 Traffic exchanges between two remote peers. 76
4.5 Distribution of the initial TTL value of the IP datagrams. 78
4.6 Distance in number of hops between peers 79
4.7 Traffic by ASes 81
4.8 Traffic by countries 81
4.9 Distance in number of AS hops 83
4.10 PPSTream: geographic location of peers and traffic 90
4.11 TVUPlayer: geographic location of peers and traffic 91
4.12 SOPCast: geographic location of peers and traffic 92
4.13 TVAnts: geographic location of peers and traffic 93
4.14 Dynamics of the traffic for PPStream and TVUPlayer 96
4.15 Dynamics of the traffic for SOPCast and TVAnts 97
4.16 Chunk distribution for BitTorrent-like P2P systems 98
4.17 New incentive mechanism for P2P live multimedia streaming 99

B.1 Client-serveur 117
B.2 CDN serveurs 118
B.3 Multicast 119
B.4 Pair-à-Pair 119
B.5 Approche *Single-tree* 121
B.6 Départ d'un pair dans un arbre 122
B.7 Approche multi-arbres 123
B.8 Architectures en maillage 124
B.9 Plateforme expérimentale de mesures 129
B.10 Débits pour TVAnts 130
B.11 Taille moyenne vs. durée des sessions 131
B.12 Octets envoyés vs. Octets reçus 135
B.13 Video download policies 136
B.14 Voisinage des pairs pour toutes les applications. 137
B.15 Durée de vide des pairs 138
B.16 Plateforme expérimentale de mesures en France et au Japon 141
B.17 Proportion de trafic (Upload/Download) 144
B.18 Échange de trafic entre différents pairs. 145
B.19 Distance en nombre de sauts des pairs sur l'Internet (CDF). 147
B.20 Volume de trafic par AS 149
B.21 Nombre de sauts d'AS 150
B.22 Localisation géographique des pairs 151

List of Tables

2.1 Architectures Summary 28

3.1 Summary of packet traces 35
3.2 Observations Summary for Traffic Patterns 38
3.3 Utilized Port number 39
3.4 Signaling Traffic Ratio 42
3.5 Number of peers interacting our host 49
3.6 CCDF of the video peers lifetime and average peers lifetime 54

4.1 Traces Summary 69
4.2 Number of peers in each trace. 72
4.3 Amount of traffic for the top three IPs 75
4.4 Default TTL value for common operating systems. 77
4.5 Amount of Traffic for the top three Ases 82
4.6 Amount of Traffic for the top three Countries 82

B.1 Récapitulatif des Architectures 126
B.2 Résumé des traces. 129
B.3 Récapitulatif du trafic observé 132
B.4 Ports utilisés 133
B.5 Proportion de trafic de signalisation. 133
B.6 Nombre de pairs 136
B.7 CCDF et durée de vie moyenne 138

Peer-to-Peer Video Live Streaming:
Measurement Experiments and Traffic Analysis

Abstract: The Internet is currently experiencing one of the most important challenges in terms of content distribution since its first uses as a medium for content delivery: users from passive downloaders and browsers are moving towards content producers and publishers. Following this change, the content delivery architecture of the Internet is also evolving from the classical client/server model to the peer-to-peer model (P2P). For new content delivery services that target a large number of receivers on the Internet, there are scalability issues to deliver the content to potentially millions of users at the same time. This is clearly the case for television service on the Internet.

The television broadcast service with a peer-to-peer architecture (P2P-TV) has became an important research topic since it is expected that P2P-TV applications will become massively used on the Internet. There has been a lot of P2P-TV architecture proposals but, none of these proposals has been really implemented or deployed. At the same time, many commercial applications appeared on the Internet (PPLive, SOPCast) and their popularity is increasing. Even tough these applications are freely available, their source code is not open and their exact implementation details and protocols are still widely unknown. There is a lack of knowledge with these applications, with regards to their architectures, mechanisms or traffic.

In this thesis, in order to fill the gap between the architecture proposals and the commercial applications, we study the P2P-TV applications by performing extensive measurement experiments. We passively measured the network traffic generated by popular P2P-TV applications.

We present a detailed study of the P2P-TV traffic, providing useful insights on both transport and packet-level properties as well as on the behavior of the peers inside the network. The knowledge gained thanks to this traffic measurement and analysis is useful for several tasks as traffic identification, understanding the performance of different P2P-TV technologies, the impact of such traffic on the network and building more realistic models for simulations.

In order to extend the scope of our study, we performed another measurement campaign at larger scale between Japan and France to better characterize the P2P network. From these experiments, we study the global organization of peers in the network, the amount of traffic they exchange and the overall collaboration of peers. We will also focus on the distance between peers and the geographic location of the users.

Keywords: content distribution, peer-to-peer, P2P-TV, measurement experiments, traffic analysis, distributed systems.
