Traffic analysis of peer-to-peer IPTV communities

Thomas Silverston [a], Olivier Fourmaux [a], Alessio Botta [b], Alberto Dainotti [b], Antonio Pescapé [b], Giorgio Ventre [b,*], Kavé Salamatian [c]

[a] UPMC Univ Paris 06, UMR 7606, LIP6/CNRS, F-75005, Paris, France
[b] University of Napoli Federico II, Department of Computer Engineering and Systems, Via Claudio 21, I-80125 Napoli, NA, Italy
[c] Lancaster University, InfoLab21, LA1 4WA, United Kingdom

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. They often distribute and retrieve multimedia contents establishing network communities. This is the case of peer-to-peer IPTV communities.

In this work we present a detailed study of P2P IPTV traffic, providing useful insights on both transport- and packet-level properties as well as on the behavior of the peers inside the network. In particular, we provide novel results on the (i) ports and protocols used; (ii) differences between signaling and video traffic; (iii) behavior of the traffic at different time scales; (iv) differences between TCP and UDP traffic; (v) traffic generated and received by peers; (vi) peers neighborhood and session duration. The knowledge gained thanks to this analysis is useful for several tasks, e.g. traffic identification, understanding the performance of different P2P IPTV technologies and the impact of such traffic on network nodes and links, and building more realistic models for simulations.[1]

1. Introduction and motivation

In recent years we are experiencing a dramatic change in how users influence the evolution of the Internet and its services. Users create events, making new content and services available; they create communities, in which active participation, user interaction, and information sharing, are highly encouraged; and demand new technologies supporting them. User demands and new forms of interaction drive the network evolution, bringing 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, etc. [1,2].

Therefore, by interacting through the network, users create new forms of communities and new forms of content distribution: we are assisting to a shift from the traditional distribution paradigm of few content providers *vs* many consumers, to a new paradigm that sees many content providers and 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 to use the Internet for activities previously happening only in certain contexts and through different technologies. This is the case, for example, of peer-to-peer IP Television (P2P IPTV), and network gaming in virtual worlds. The time and place of such activities change, and services become ubiquitous. People move from the *sofa at home* to the workplace or a café to enjoy

* Corresponding author. Tel.: +39 081 7683908; fax: +39 081 7683816.
E-mail addresses: thomas.silverston@lip6.fr (T. Silverston), olivier.fourmaux@lip6.fr (O. Fourmaux), a.botta@unina.it (A. Botta), alberto@unina.it (A. Dainotti), pescape@unina.it (A. Pescapé), giorgio.ventre@unina.it (G. Ventre), kave.salamatian@lancs.ac.uk (K. Salamatian).

[1] The work described in this paper has been funded by the CONTENT EU Network of Excellence (IST-FP6-038423).

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.

The popularity of Internet-based television is expected to grow during the next years for several reasons [8]. 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 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) [6]. Third, in some countries the quality and the range of the offer of TV contents is scarce. Finally, as fourth, the "*Broadcast yourself*" phenomenon is constantly increasing, both with "**Tube*" sites and the creation of more elaborate TV programs with realtime broadcasting created by single users [7,4,5]. To testify such trends, several sources report on the loss of audience and of monetary income of the traditional TV industry. Therefore, the interest in understanding such new technologies to support and improve them is enormous [7].

The use of the P2P paradigm to deliver live television on the Internet (P2P IPTV) is gaining increasing attention [9], and has become a promising alternative to other legitimate approaches as the classical client–server model, content delivery networks (CDNs) [61], or IP-Multicast. Indeed, television service targets a large number of users and a simple client–server approach will not scale to a large audience because servers have limited available resources (CPU, bandwidth) that will decrease proportionally with the number of users. By multiplying the servers, CDNs only scale to a larger audience with regards to the number of deployed servers. CDNs have also a high infrastructure cost, which will partially limit its use by the content providers. Finally, the lack of deployment of IP-Multicast limits the availability and scope of this approach for a TV service on the Internet scale [62]. In P2P networks, instead, peers will contribute their resources (CPU, upload bandwidth) and are at the same time downloaders and uploaders of realtime video-streams. The available resources to deliver the content increase with the number of users and can scale to a large user population, without any additional infrastructure cost. Moreover, by using the existing Internet infrastructure as a medium and by exploiting user participation for the creation of the content distribution network, P2P IPTV technologies have innovative potentials: (i) to make any TV channel from any country globally available, (ii) to make each Internet user a content creator and distributor by broadcasting his own "TV" with trivial costs. These are some of the reasons behind the increasing popularity of such applications among Internet users. This trend is also confirmed by the amount of new P2P IPTV applications that become continuously available, and by the fact that the traffic generated by such applications has recently increased significantly.

In this paper we point our attention on the study of P2P IPTV communities. More precisely, we study the traffic generated by the four most used P2P IPTV applications at the time of the experiment, and still considered today among the top P2P IPTV applications: *PPLive*, *PPStream*, *Sopcast*, *TVants*. Analyzing four applications instead of a single one makes our analysis more complete and allows to investigate the generalizability of the observed results. One of the contexts that have brought P2P IPTV to the attention of Internet users and have also pushed new people to use the network and participate to network communities, is that of worldwide sport events. Such applications allowed people from all over the world to watch events not broadcast (or not freely broadcast) by their national TVs. For this reason, in this paper we chose to analyze the traffic generated by peers of the community watching the 2006 FIFA World Cup (June/July 2006).

The work here aims 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. We aim at understanding: (i) which transport-level protocols are used and what are the consequences of different choices; (ii) how traffic is divided into signaling and data, and into upload and download directions, in order to study and characterize them separately; (iii) criteria useful to discriminate between signaling and data traffic and to identify P2P IPTV traffic; (iv) statistical properties of P2P IPTV useful to understand the impact on network nodes and links (e.g. long-range dependence); (v) how peers interact, how much they contribute to the content distribution, and what is their typical lifetime; and (vi) what is the download policy of the different applications. The results presented here are relevant to identify traffic generated by such applications, to understand their impact on network nodes and links, and to build realistic simulations and emulations.

The paper is structured as follows: we describe the considered applications and the measurement setup in Section 2. Afterward, we analyze the results related to lower-level traffic characteristics in Section 3, and those related to peers behavior in Section 4. In Section 5 we overview the literature related to the measurement of P2P IPTV communities. Finally, Section 6 ends the paper with discussion and conclusion remarks.

2. Description of the experiments

With the aim to better understand both traffic properties and peer behavior of a P2P IPTV community during a worldwide event, we considered four applications. Analyzing different applications allows studying such communities 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 IPTV 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 [56]. Some of the strongest motivations for people to resort to P2P IPTV 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.

2.1. Studied P2P IPTV applications

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 [11], 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 [18], estimates indicate millions of concurrent users.

All the largely deployed P2P IPTV systems claim to use a mesh-based architecture as those investigated in this paper. The mesh-based architecture used by P2P IPTV systems takes its inspiration from BitTorrent [12] and uses the same kind of swarming protocol, as in Donet [39]. 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. With no static topology, the meshed-based architecture is more suited to deal with the peer churn than the previously proposed tree-based architecture [13,14]. Strict topologies like tree were very sensitive to the churn of peers, that is, when peers are prone to failures or may eventually leave the network, which is a frequent behavior in P2P networks [15]. Moreover, several studies show that the mesh-based architecture outperforms the tree-based architecture [16,17].

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 with each other a *buffer map* representing the data they have. 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 [40]. Thus, these P2P protocols generate 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 to download the video data chunks by exchanging randomly with other peers information about the data chunks they have (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 developers are to use a mesh-based architecture, their source code is not open and their exact implementation details and protocols are still widely unknown. Therefore, we can only count on traffic analysis to understand their transmission mechanisms and peer behavior.

2.2. 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. In this paper we focus on four packet traces, one for each application, collected on June 30 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 IPTV users. The traces are publicly available at [29]. It is worth stating that we also analyzed the other collected traces and we obtained results similar to those presented in this paper.

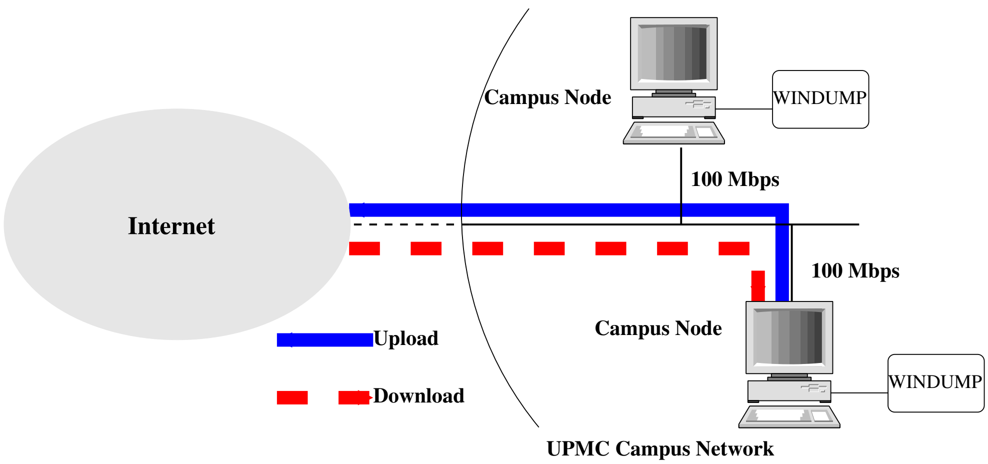


Fig. 1. Measurement experiments testbed. Each node is a common PC directly connected to the Internet via campus network.

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: 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 IPTV application as well as `WinDump` [58] to collect the traffic. Therefore we collected two traffic traces for each match, one for each application. In particular, we respectively collected traffic from PPStream and SOPCast during the the first match and from PPLive and TVAnts during the second one.

Our measurement testbed is described in Fig. 1. To collect packets, we used two PCs equipped with 1.8 GHz 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 100 Mbps `Ethernet` link. For all the measurement experiments, the consumed bandwidth was always relatively low and did not exceed 10 Mbps. 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 assure that the behavior of peers 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 was 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. This was done both manually and using Plab [57], a software for traffic analysis at packet-level that we also used to obtain packet-level, flow-level, and host-level measures used in this paper.

3. Understanding P2P IPTV 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 [46].

3.1. Protocols and ports

The considered applications generate traffic using different ports and protocols. Table 1 contains the information regarding the used protocols and the sizes of the traces. The time duration of the collection (≈225 min) is longer than that of a soccer match (≈105 min). 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 behavior of peers (e.g. flash crowds).

Table 1
Summary of packet traces.

	PPLive	PPStream	SOPCast	TVAnts
Duration (s)	13,321	12,375	12,198	13,358
Size (MB)	6339	4121	5475	3992
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

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 Fig. 1) than in the download one (i.e. from all the other peers to our node, red dashed line in Fig. 1). 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. Interestingly, we can notice that PPLive, TVAnts and PPStream make extensive use of TCP, whereas SOPCast runs mainly on UDP. Moreover we can observe that TVAnts also relies on UDP for a non negligible percentage of packets.

Table 2 shows the ports used by the applications. PPLive and SOPCast present a similar behavior. Indeed, with for 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 16,800, both local and remote, for most of the UDP and TCP packets. This is probably because TVAnts sets a default port on a new installation that can be changed thereafter by the user. Looking at Table 2, it is also evident how P2P IPTV traffic cannot be reliably identified by looking at

Table 2
Utilized port number (percentage of packets).

		PPLive	PPStream	SOPCast	TVAnts
Remote peers	TCP	Several	Several	Several	16,800 (>25%)
	UDP	Several	7201 (100%)	Several	16,800 (>60%)
Controlled peer	TCP	10,549 (>99%)	11,430 (>99%)	8516 (>99%)	16,800 (>71%)
	UDP	5747 (100%)	5747 (42%), 11,430 (54%), 65,535 (4%)	8516 (>99%)	16,800 (>99%)

transport protocol ports, motivating the need to find different ways to recognize their traffic.

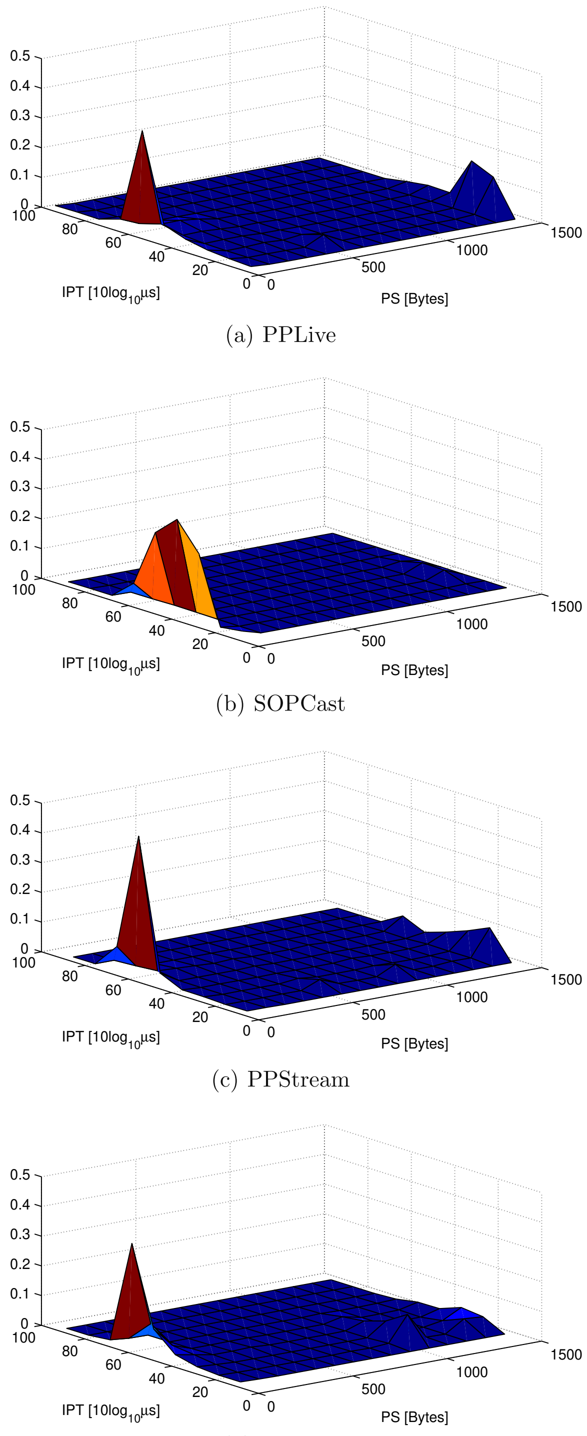
3.2. Signaling and video traffic

As we explained in Section 2, the P2P applications we studied generate two kinds of traffic: video and signaling. The signaling traffic of P2P IPTV systems is not expected to be delay-sensitive, because it is used for exchanging information about peers or data availability but not for interactive commands, as for video on-demand systems like Joost [34]. In video on-demand systems, the users may want to move the video playback instant forward or backward promptly. In the case of P2P IPTV, it is not possible to have this kind of interactive commands since the data flows are broadcast live. In general we can say that the signaling and 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 [33]. 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.

Figs. 2 and 3 reveal interesting properties of overall P2P IPTV 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.

Fig. 2 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.



(a) PPLive

(b) SOPCast

(c) PPStream

(d) TVAnts

Fig. 2. Joint probability distribution of inter packet time and packet size.

In Fig. 3 instead, we show scatter plots in which, the coordinates of each point are given by the average PS and the number of transmitted packets of each session. The

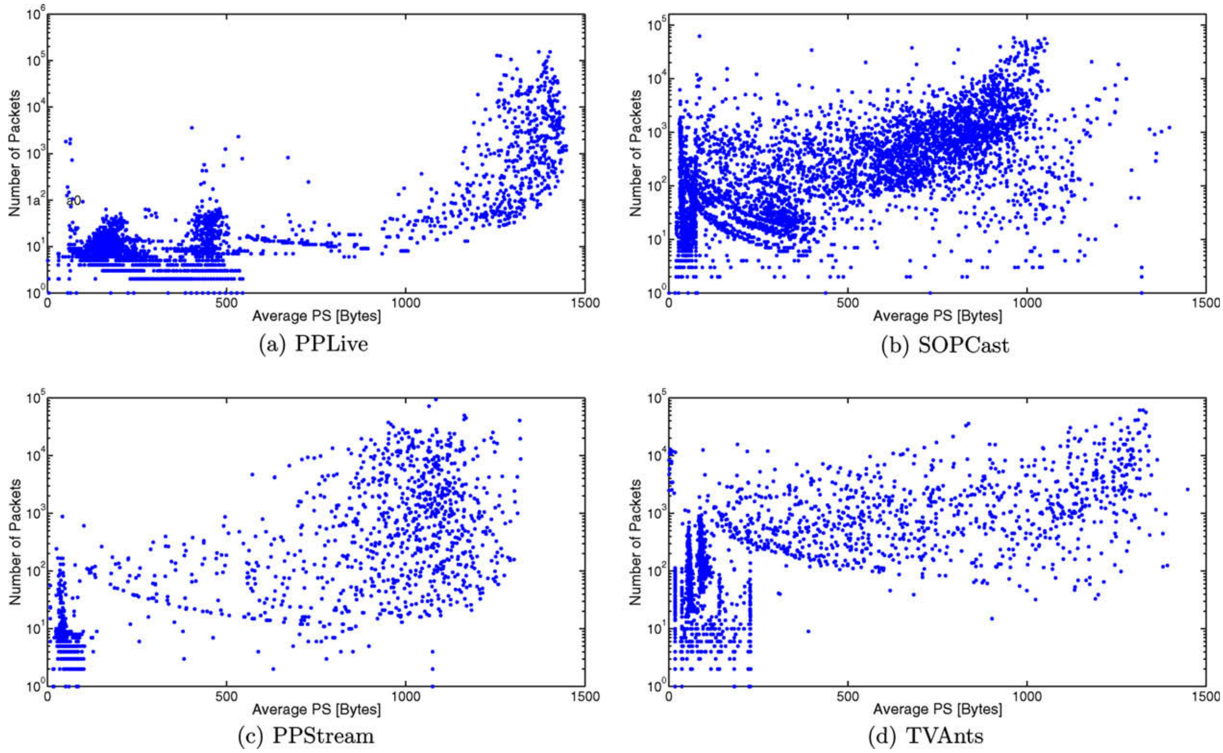


Fig. 3. Upstream flows: average packet size vs number of packets.

number of transmitted packet 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 P2P IPTV applications considered. 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. Finally, 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 we report statistics on the ratio of signaling traffic with respect to overall traffic of all the applications, also separated in download and upload. We observe 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.

Table 3
Signaling traffic ratio.

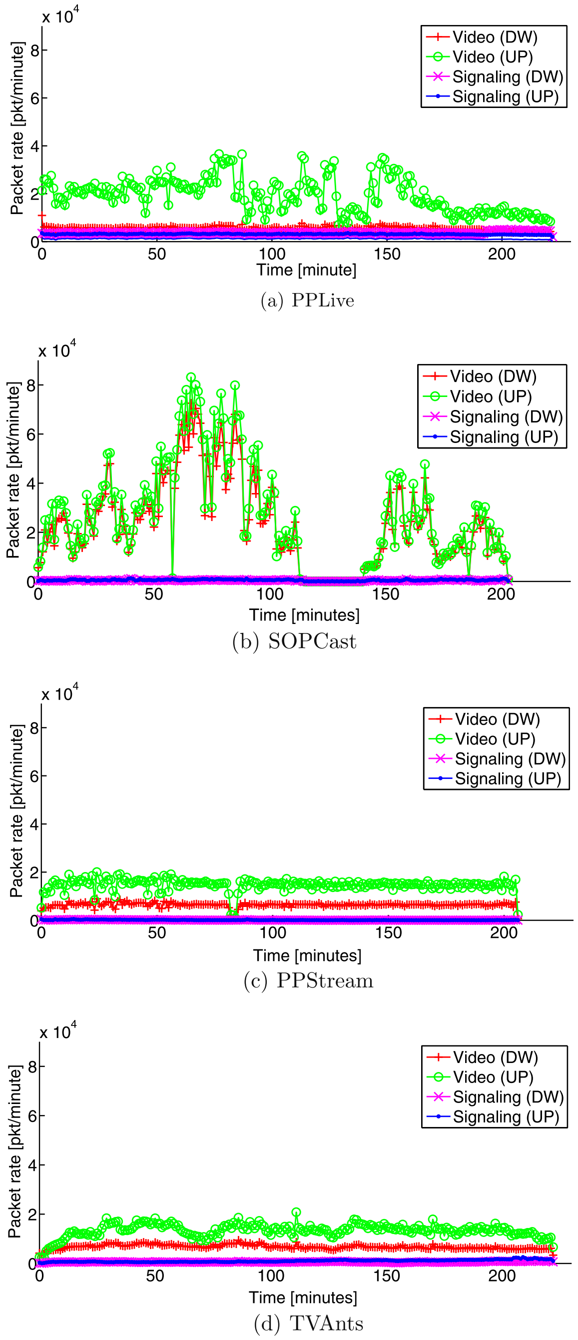
	PPLive	PPStream	SOPCast	TVAnts
Total (%)	4.1	13.6	19.3	10.2
Upload (%)	2.2	10.8	13.6	7.8
Download (%)	19.2	25.8	48.5	18.0

Looking at the packet rate for each of them and for both upload and download directions, Fig. 4 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 observe that SOPCast generates a packet rate higher than all the other applications, 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. Therefore, we attribute this behavior to the main source of content and not to the network.[2]

3.3. Scaling behavior

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 [30]. The smallest time scale we consider is related to 20 ms intervals, as we observed from

[2] 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 other three applications.



(a) PPLive

(b) SOPCast

(c) PPStream

(d) TVAnts

Fig. 4. Packet rate of video and signaling traffic in download and upload directions (bin duration is 60 s).

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) [31], 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 $t = 2^j * 20$ ms. 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. More details about the scaling analysis of P2P ITPV traffic are reported in [32].

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 3.2). Therefore, for each application, we obtained four distinct plots: overall upload traffic, video upload traffic, overall download traffic and video download traffic.

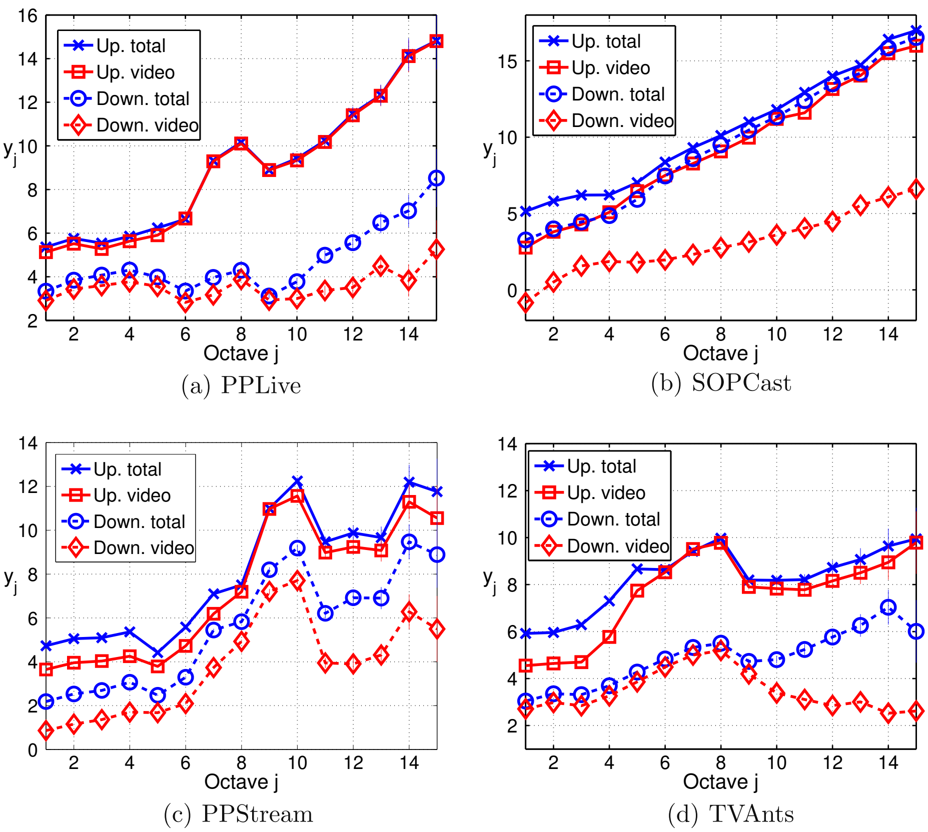
Fig. 5a–d present the energy spectra on a logscale graph for PPlive, SOPCast, PPStream and TVAnts, respectively.

As shown in Table 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.

3.3.1. 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), Fig. 5b shows that the four energy spectra look similar for all the traffic directions and kinds (the slight difference for the video download energy spectra, dashed line with rhomboidal markers, 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. $2^8 * 20$ ms $= 5.12$ s). 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. 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



(a) PPLive (b) SOPCast (c) PPStream (d) TVAnts

Fig. 5. Traffic energy spectra (bin width is 20 ms).

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 [46]. This also illustrates that the P2P IPTV 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 the not so trivial choice of transport protocols for P2P IPTV systems. 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.

3.3.2. 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 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 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, 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.

4. Understanding P2P IPTV 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 IPTV communities.

4.1. 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 4 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.

Fig. 6 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 viceversa. In particular, a point over the bisector (bold solid line in Fig. 6) represents a peer that received more data than it sent, while a point under the bisector is representative of a peer that 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 with which our host interacts 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

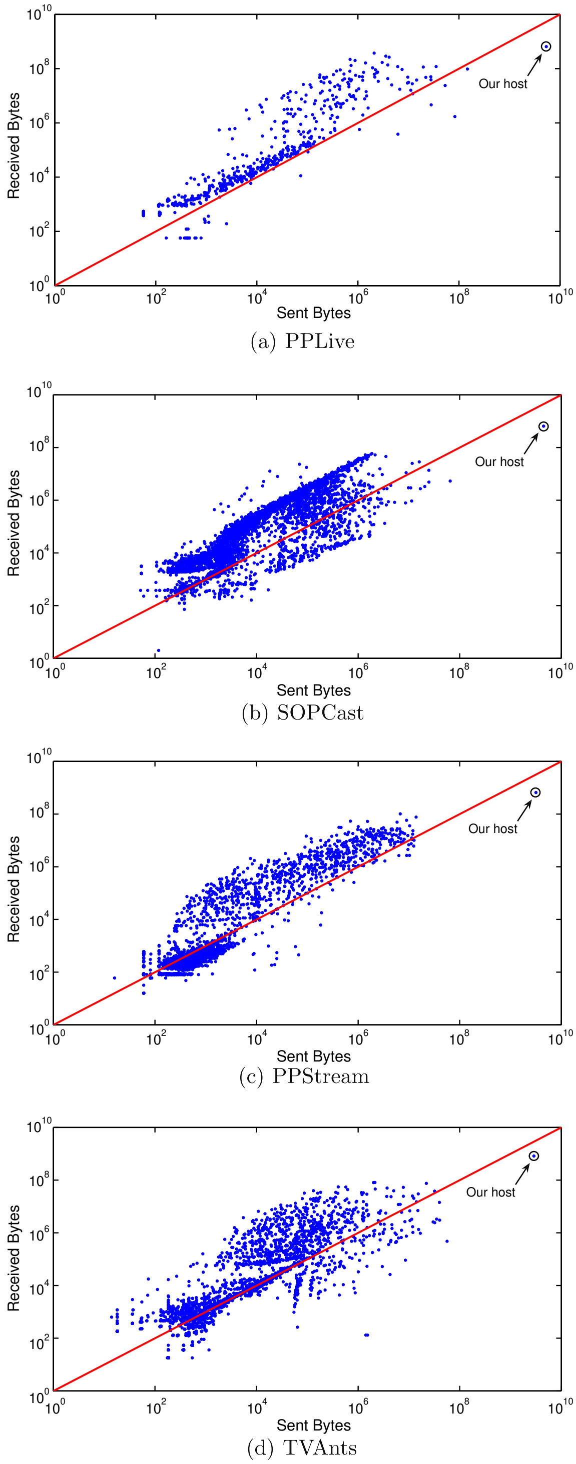


Fig. 6. Bytes sent vs Bytes received for each host.

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 Fig. 6 related to our host is located under the bisector. Moreover, it is interesting to note that a large quantity of peers are in the lower-

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

	PPLive	PPStream	SOPCast	TVAnts
Total	649	5956	3876	5394
Non cooperative region	305	3186	3012	4075
Percentage (%)	47	53.5	77.7	75.5

left region of the diagrams (<10 KB in both directions), meaning that their interaction with our host is very low. In Table 4, to help to interpret the diagrams in Fig. 6, we report the percentage of peers inside the $(0,0)$–$(10^4, 10^4)$ rectangle, which we call the *"Non cooperative region"*. Such percentages are quite high, showing that from about 50–75% 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 sort of instability and overhead. It is also worth noticing that the worst results are related to the two applications using UDP.

Another interesting aspect of such graphs is that, in general, there seems to be a sort of proportion between what the peers send and receive: the points in the graphs are not very sparse and a straight envelope can be easily recognized for the vast majority of the peers. This may indicate both that: (i) the relations of proximity that the peers have with our host (e.g. delay) and their access-links available bandwidth affect their behavior in terms of quantity of data exchanged; (ii) the applications try to keep a proportion between inbound and outbound throughput. This observation regarding P2P IPTV systems, which are BitTorrent-like systems, recalls a similar finding in [47], where Legout et al. observed a "clustering of similar-bandwidth peers" in the BitTorrent system.

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 Fig. 7 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 s interval (i.e. bin duration is 60 s).

As said, SOPCast (Fig. 7c) received no traffic from 130 to 140 min, 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, probably the video source has suffered technical problems. Fig. 7 shows that the download policies for all the applications are different. For PPLive (Fig. 7a), 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 (Fig. 7b) 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 (Fig. 7c) 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.

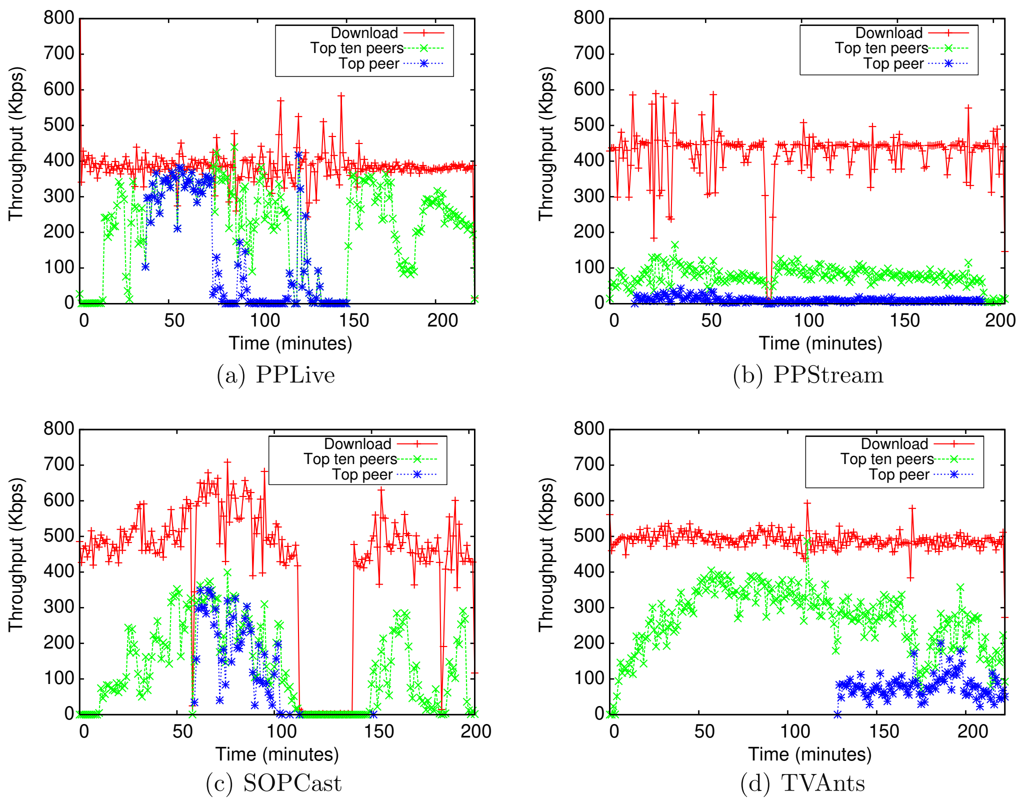


Fig. 7. Video download policies: total traffic, top-ten peers traffic and top-peer traffic (bin duration is 60 s).

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 (Fig. 7d) seems to mix PPStream and SOPCast 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.

4.2. *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 Fig. 8 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 corroborates 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 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 IPTV, 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 IPTV systems have to deal with the arrivals and departures of peers (i.e. *churn* of peer). This is a challenge because live video has to respect playout point to achieve smooth rendering. A high churn of peers will involve additional delays or jitters for packet delivery, which will decrease overall video quality. Here we show the video-peer lifetime to point out the churn of peers. Since our nodes have only a local view of all the peers in the network, the video-peer lifetime is the duration between the first and the last time our controlled nodes exchange video traffic with another peer. As a representative example, Fig. 9 plots the complementary cumulative distribution function (CCDF) of TVAnts video peer lifetime. It follows a Weibull distribution. This applies to all the four applications (the CCDF plots for the other applications can be found in [35]). The parameters of the Weibull distribution functions we used for fitting the measured video-peer lifetime are presented in Table 5. Such table also shows the average peer lifetime.

For all the applications, there are no more than 10% of peers that stay in the network during an entire match. Moreover, the average video-peer lifetime is different for all the applications and it is far from an entire match dura-

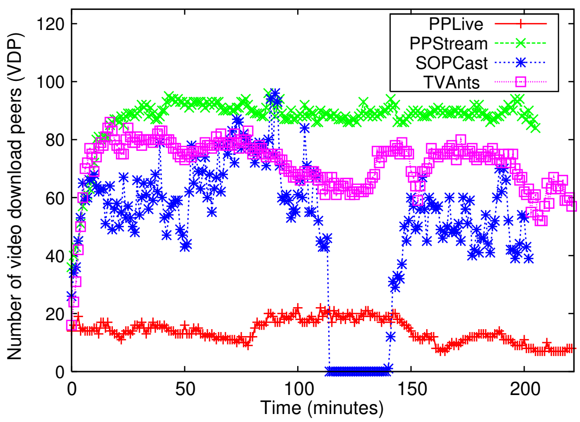


Fig. 8. Peers neighborhood for all the applications (bin duration is 60 s).

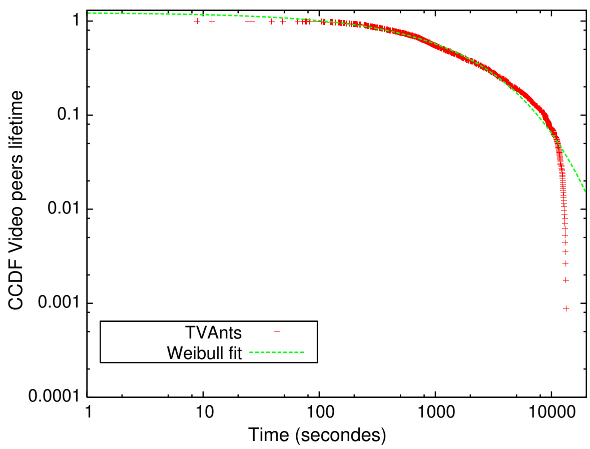


Fig. 9. Video-peers lifetime for TVAnts.

Table 5
Video-peers lifetime summary.

	Video lifetime complementary CDF	Avg. peer lifetime (s)
PPLive	$2.0 * e^{-(x/12.3)^{0.2}}$	393
PPStream	$1.2 * e^{-(x/322.1)^{0.4}}$	1222
SOPCast	$1.1 * e^{-(x/993.8)^{0.4}}$	1861
TVAnts	$1.2 * e^{-(x/1572.8)^{0.6}}$	2778

tion. The departure of a peer can be due to a user that stops watching the game or due to the application mechanisms which force switching from a video peer to another one. Since all the applications exhibit a Weibull distribution for video-peers lifetime, our understanding is that Weibull distributions are driven by user behavior.

5. Related work

As introduced in Section 1, P2P IPTV is a user-driven evolution of network applications, often involving community-based organization of peers. For example, in the Chinese communities all over the world, the annual Spring Festival Gala on Chinese New Year is one of the most popular TV programs. As reported in [33], in 2006 starting from 3AM EST of January 28 (Chinese New Year Eve day), 14 PPLive channels were broadcasting the event. This is just the first example illustrating how p2p broadcast TV entails user community formation in the Internet. Another example are the P2P IPTV communities centered on worldwide sport events: web sites as [11] provide information regarding the most important events that will be broadcast on the various channels through the various applications, and at the same time provide tools for exchanging information among the users. In addition, it is worth noticing that a large part of applications for P2P IPTV (and consequently a number of users communities) were born in countries like China, where a strict control over the distribution of the contents and a strong censorship is present: all in all, socio-cultural–political issues can drive the evolution of user communities on the Internet.

For these reasons, during the last years, the research community has paid an increasing attention to measurement studies of P2P IPTV scenarios, conducted with the aim to analyze the mechanisms of such systems, the traffic profiles, the perceived quality, and the behavior of the involved peers. This also entails new measurement approaches [45].

The analysis and the characterization of P2P IPTV traffic is of indisputable interest for a large number of reasons: (i) to improve the understanding of this new traffic typology and to pave the way for identification and classification approaches; (ii) to evaluate the impact of this traffic for supporting design, planning, optimization, provisioning, and forecasting stages; (iii) being synchronous network applications, P2P IPTV applications have stringent quality of service constraints (e.g. bandwidth, delay, jitter) and their traffic characterization will enable understanding their exact needs in terms of network resources; (iv) to develop synthetic traffic generation models that can be used when modeling or simulating these systems. For instance, an important concern of P2P IPTV systems is the scalability. The traffic analysis and characterization may help to estimate the impact of overhead traffic generated by the signaling. Finally, from the application point of view, global knowledge of the traffic properties will highlight some drawbacks of the applications and will make it possible to improve the design of new P2P IPTV architectures.

Even though lots of measurement studies have been conducted on P2P file sharing [19–22] and telephony systems [23–28], very few tackled P2P IPTV. Sripanidkulchai et al. [36] showed that large-scale live streaming can be supported by P2P end-user applications despite the heterogeneous capacity of peers, paving the way to future studies in the field of P2P IPTV. Zhang et al [37] presented the first measurement results about their protocol Donet [39], which was deployed on the Internet and called Coolstreaming. They provided network statistics, understanding of the user behavior in the whole system, and results related to the quality of video reception. In [33,41] Hei et al. made a complete measurement of the popular PPLive application. They made active measurements by configuring their own crawler and providing many architecture and overlay details such as buffer size and number of peers in the networks. Based on their measurement studies, Hei et al. [40] developed a methodology to estimate the overall perceived video quality throughout the network. Vu et al. [42] made active measurements of the PPLive system and derived mathematical models for the distributions of channel population size and session length. Ali et al. [43] made passive measurements of PPLive and SOPCast applications and analyzed the performance and characteristics of such systems. Still in their previously mentioned works, Ali et al. provided their own methodology to study the data exchanges of such P2P applications. Our work is different from these, since we do not focus on a single application or, as in the case of [43], on a couple of applications, but on a set of four applications used worldwide. An important distinction between Hei works and ours comes from the live interest of the measured event. It is intuitive but corroborated by Veloso et al. [38] that traffic patterns have not the same characteristics as to whether broadcast content exhibits a live interest for users or not. In our previous work [44], we passively measured the network traffic generated by several popular applications during a worldwide event. We compared the measured applications by inferring their underlying mechanisms and highlighted their design differences and similarities. Compared to our previous work, in this paper we add (i) a deeper traffic analysis; (ii) a scaling analysis to characterize the correlation structure of the generated traffic at different time scales to understand its properties and its impact on the network; (iii) a careful analysis of peers behavior in P2P IPTV communities.

6. Discussion and conclusion

Despite many issues still being open (e.g. copyright of exchanged content [48,49], performance of IPTV over wireless networks [52,53], diffusion of broadband connections [55,59], quality of experience [50,51], standardization

[54]), P2P IPTV traffic has increased a lot and it will largely contribute to increase the overall Internet traffic [9,10].

In this work we analyzed the network traffic generated by four of the most popular P2P IPTV applications. Such applications use proprietary unpublished protocols, making their study challenging. However this work leads towards improving knowledge of current P2P IPTV systems. We think 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.

We outlined similarities and differences among such 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 IPTV traffic is to discriminate between signaling and data traffic. We discovered several properties of the traffic that strongly confirm, 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 IPTV communities. Moreover, we gained some knowledge regarding statistical properties of this traffic (e.g. PS-IPT distribution, recurring PS, etc.) that in the future we plan to further investigate as means for application identification through traffic analysis. Moreover, this study allows understanding how traffic from peers participating to a P2P IPTV 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 better understanding 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 IPTV communities. 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 realtime 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 of an increase in the overhead). Moreover, we found that, in general, the amount of data sent to each peer by our nodes was sensibly 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, deriving useful insights on peers' behavior in terms of traffic contribution and of 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 showed that their distribution can be fitted with a Weibull function. This result is probably a consequence of user behavior and not of the specific software architectures, since all the applications exhibit a Weibull distribution for video-peers lifetime. This information is useful as a reference model for simulations.

Due to the lack of realistic models for P2P traffic [60], simulations could lead to wrong results. Thanks to the results shown and discussed in this paper, 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 IPTV 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 IPTV applications and users, such communities are rapidly expanding and they should be therefore constantly monitored. For this reason our current work is concerned with understanding the differences between past and current versions of P2P IPTV applications. The analysis of their evolution should generate interesting insights regarding the possible future directions. Moreover, we are dealing with the assessment of models of both application traffic and peer behavior which can be utilized in simulation and emulation scenarios.

Acknowledgement

The authors would like to thank Lucio Tufari from University of Napoli for the interesting discussions and his valuable support on data analysis.

References

[1] M. Knights, Web 2.0, Communications Engineer 5 (1) (2007) 30–35. February–March.
[2] I.V. Yakovlev, Web 2.0: is it evolutionary or revolutionary?, IT Professional 9 (6) (2007) 43–45 November–December 2007.
[3] S. Murugesan, Understanding Web 2.0, IT Professional 9 (4) (2007) 34–41. July–August.
[4] B. Garfield, The YouTube effect, Wired Magazine (2006) 222. December.
[5] S. Boll, MultiTube-where Web 2.0 and multimedia could meet, IEEE Multimedia 4 (1) (2007) 9–13. January–March.
[6] D.H. Shin, Potential user factors driving adoption of IPTV. What are customers expecting from IPTV?, Technological Forecasting and Social Change 74 (8) (2007) 1446–1464 October.
[7] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, S. Moon, I Tube, You Tube, Everybody Tubes: analyzing the world's largest user generated content video system, in: ACM SIGCOMM/USENIX Internet Measurement Conference (IMC), 2007.
[8] R. Jain, I want my IPTV, Multimedia, IEEE 12 (3) (2005) 96. July–September.
[9] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, S. Tewari, Will IPTV ride the peer-to-peer stream? (peer-to-peer multimedia streaming), IEEE Communications Magazine 45 (6) (2007) 86–92. June.
[10] Y. Xiao, X. Du, J. Zhang, Fei Hu, S. Guizani, Internet protocol television (IPTV): the killer application for the next-generation internet, IEEE Communications Magazine 45 (11) (2007) 126–134. November.
[11] http://myp2p.eu (April 2008).

[12] B. Cohen, Incentives build robustness in bittorrent, in: Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA, May 2003.
[13] M. Bawa, H. Deshpande, H. Garcia-Molina, Transience of peers and streaming media, ACM SIGCOMM Computer Communication Review 33 (1) (2003) 107–112.
[14] V. Venkatraman, K. Yoshida, P. Francis, Chunkyspread: heterogeneous unstructured end system multicast, in: Proceedings of the IEEE ICNP, 2006.
[15] D. Stutzbach, R. Rejaie, Understanding churn in peer-to-peer networks, in: Proceedings of the ACM SIGCOMM/USENIX Internet Measurement Conference, 2006.
[16] T. Silverston, O. Fourmaux, Source vs data-driven approach for live P2P streaming, in: Proceedings of the IEEE International Conference on Networks, 2006.
[17] N. Magharei, R. Rejaie, Mesh or multiple-tree: a comparative study of live P2P streaming approaches, in: Proceedings of the IEEE Infocom, 2007.
[18] Y. Huang, T.Z.J. Fu, D.-M. Chiu, J.C.S. Lui, C. Huang, Challenges, design and analysis of a large-scale P2P-VoD system, in: Proceedings of the ACM SIGCOMM, 2008.
[19] K.P. Gummadi, R.J. Dunn, S. Saroiu, S.D. Gribble, H.M. Levy, J. Zahorjan, Measurement, modeling, and analysis of a peer-to-peer file-sharing workload, in: Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19), Bolton Landing, New York, October 2003.
[20] K. Tutschku, A measurement-based traffic profile of the eDonkey filesharing service, in: Proceedings of the Fifth Annual Passive and Active Measurement Workshop (PAM'04), Antibes Juanles-Pins, France, April 2004, pp. 12–21.
[21] M. Zhou, Y. Dai, X. Li, A measurement study of the structured overlay network in P2P file-sharing applications, in: Proceedings of the Eighth IEEE International Symposium on Multimedia, December 11–13, 2006, pp. 621–628.
[22] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips, The BitTorrent P2P file-sharing system: measurements and analysis, in: Proceedings of The Fourth International Workshop on Peer-to-Peer Systems (IPTPS'05), Ithaca, NY, February 2005.
[23] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, P. Tofanelli, Revealing skype traffic: when randomness plays with you, ACM SIGCOMM Computer Communication Review 37 (2007) 37–48. October.
[24] D. Rossi, M. Mellia, M. Meo, A detailed measurement of Skype network traffic, in: Proceedings of Seventh International Workshop on Peer-to-Peer Systems (IPTPS'08), Tampa Bay, Florida, 25–26/2/2008.
[25] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, S. Valenti, P. Veglia, Pictures from the Skype, in: Proceedings of the ACM Sigmetrics 2008, Annapolis, Maryland, USA, June 2–6, 2008.
[26] D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca, D. Rossi, Tracking down Skype traffic, in: Proceedings of the IEEE Infocom, Phoenix, AZ, 15,17 April 2008.
[27] H. Xie, Y.R. Yang, A measurement-based study of the Skype peer-to-peer VOIP performance, in: Proceedings of Sixth International Workshop on Peer-to-Peer Systems (IPTPS'07).
[28] K. Suh, D.R. Figueiredo, J. Kurose, D. Towsley, Characterizing and detecting Skype-relayed traffic, in: Proceedings of 25th IEEE International Conference on Computer Communications, INFOCOM 2006, April 2006, pp. 1–12.
[29] http://content.lip6.fr/traces/ (April 2008).
[30] P. Abry, D. Veitch, A wavelet-based joint estimator of the parameters of long-range dependence, Transactions on Information Theory, IEEE 45 (3) (1999) 878–897. April.
[31] D. Veitch, P. Abry, Matlab code for the wavelet based analysis of scaling processes (Online). Available: http://www.cubinlab.ee.mu.oz.au/darryl (April 2008).
[32] T. Silverston, O. Fourmaux, K. Salamatian, Characterization of P2P IPTV traffic: scaling analysis, http://arxiv.org/abs/0704.3228, TR 2007.
[33] X. Hei, C. Liang, J. Liang, Y. Liu, K.W. Ross, Insights into pplive: a measurement study of a large-scale p2p IPTV system, in: Proceedings of the IPTV Workshop, International World Wide Web Conference, 2006.
[34] http://www.joost.com (April 2008).
[35] T. Silverston, O. Fourmaux, P2P IPTV measurement: a comparison study, http://www.arxiv.org/abs/cs.NI/0610133, TR 2006.
[36] K. Sripanidkulchai, A. Ganjam, B. Maggs, H. Zhang, The feasibility of supporting large-scale live streaming applications with dynamic application end-points, in: Proceedings of the ACM SIGCOMM, 2004.
[37] X. Zhang, J. Liu, B. Li, On large-scale peer-to-peer live video distribution: coolstreaming and its preliminary experimental results, in: Proceedings of the MMSP, 2005.
[38] E. Veloso, V. Almeida, W. Meira, A. Bestavros, S. Jin, A hierarchical characterization of a live streaming media workload, in: Proceedings of the ACM SIGCOMM IMW, 2002.
[39] X. Zhang, J. Liu, B. Li, T.P. Yum, Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming, in: Proceedings of the IEEE Infocom, 2005.
[40] X. Hei, Y. Liu, K.W. Ross, Inferring network-wide quality in p2p live streaming systems, IEEE JSAC (2007). special issue on P2P Streaming, December.
[41] X. Hei, C. Liang, J. Liang, Y. Liu, K.W. Ross, A measurement study of a large-scale p2p IPTV system, IEEE Transactions on Multimedia (2007). December.
[42] L. Vu, I. Gupta, J. Liang, K. Nahrstedt, Measurement and modeling of a large-scale overlay for multimedia streaming, in: Proceedings of the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, QSHINE'07, 2007.
[43] S. Ali, A. Mathur, H. Zhang, Measurement of commercial peer-to-peer live video streaming, in: Proceedings of the Workshop in Recent Advances in Peer-to-Peer Streaming, 2006.
[44] T. Silverston, O. Fourmaux, Measuring p2p IPTV systems, in: Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV'07, 2007.
[45] D. Ciullo, M. Mellia, M. Meo, Traditional IP measurements: what changes in a today multimedia IP network, in: Proceedings of the Fourth International Telecommunication Networking Workshop on QoS in Multiservice IP Networks, 2008, IT-NEWS 2008, 13–15 February 2008, pp. 262–267.
[46] A. Erramilli, O. Narayan, W. Willinger, Experimental queueing analysis with long-range dependent packet traffic, IEEE/ACM Transactions on Networking 4 (2) (1996) 209–223.
[47] A. Legout, N. Liogkas, E. Kohler, L. Zhang, Clustering and sharing incentives in BitTorrent systems, in: Proceedings of the ACM Sigmetrics 2007, San Diego, June 12–16, 2007.
[48] Y. Guo, C. Lin, H. Yin, Z. Zhao, Design and analysis of IPTV digital copyright management security protocol, in: Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems, 2007, ISPACS 2007, November 28, 2007–December 1, 2007, pp. 554–557.
[49] Z. Hua, C. Chunxiao, Z. Li, Y. Shiqiang, Z. Lizhu, Content protection for IPTV-current state of the art and challenges, in: Proceedings of the IMACS Multiconference on Computational Engineering in Systems Applications, 4–6 October 2006, pp. 1680–1685.
[50] J. Kishigami, The role of QoE on IPTV services style, in: Proceedings of the Ninth IEEE International Symposium on Multimedia, 2007, ISM 2007, 10–12 December 2007, pp. 11–13.
[51] M.N. Zapater, G. Bressan, A proposed approach for quality of experience assurance of IPTV, in: Proceedings of the First International Conference on Digital Society, 2007, ICDS'07, January 2007, pp. 25–25.
[52] H. Singh, K. ChangYeul, S.S. Kim, C. Ngo, IPTV over WirelessLAN: promises and challenges, in: Proceedings of the Fifth IEEE Consumer Communications and Networking Conference, 2008, CCNC 2008, 10–12 January 2008, pp. 626–631.
[53] F.E. Retnasothie, M.K. Ozdemir, T. Yucek, H. Celebi, J. Zhang, R. Muththaiah, Wireless IPTV over WiMAX: challenges and applications, in: Proceedings of the IEEE Annual Wireless and Microwave Technology Conference, 2006, WAMICON'06, 4–5 December 2006, pp. 1–5.
[54] C.-S. Lee, IPTV over next generation networks in ITU-T, in: Proceedings of the Second IEEE/IFIP International Workshop on Broadband Convergence Networks, 2007, BcN'07, 21–21 May 2007, pp. 1–18.
[55] J.E. Simsarian, M. Duelk, IPTV bandwidth demands in metropolitan area networks, in: Proceedings of the 15th IEEE Workshop on Local and Metropolitan Area Networks, 2007, LANMAN 2007, 10–13 June 2007, pp. 31–36.
[56] http://www.mobilemarketingmagazine.co.uk/mobile_tv/index.html (April 2008).
[57] http://www.grid.unina.it/software/Plab (April 2008).
[58] http://windump.polito.it (April 2008).
[59] G. Greg, Redefining the server as home networks emerge, Internet Computing, IEEE 12 (2) (2008) 7–9.
[60] C. Rolland, J. Ridoux, B. Baynat LiTGen, a lightweight traffic generator: application to P2P and mail wireless traffic, in: Proceedings of the Eighth Annual Passive and Active Measurement Workshop (PAM'07), Louvain-la-neuve, Belgium, April 2007.
[61] http://www.akamai.com (April 2008).
[62] C. Diot, B.N. Levine, B. Lyles, H. Kassem, D. Balensiefen, Deployment issues for the IP multicast service and architecture, IEEE Network 14 (1) (2000) 78–88.