Measuring P2P IPTV Systems

Thomas Silverston
Université Pierre et Marie Curie - Paris 6
Laboratoire LIP6/CNRS, UMR 7606
104 avenue du président Kennedy
75016 Paris, France
thomas.silverston@lip6.fr

Olivier Fourmaux
Université Pierre et Marie Curie - Paris 6
Laboratoire LIP6/CNRS, UMR 7606
104 avenue du président Kennedy
75016 Paris, France
olivier.fourmaux@lip6.fr

ABSTRACT

P2P IPTV systems start to be very popular on the Internet. Measuring the impact they have on the network and understanding their behavior is an important field. Available applications are based on proprietary solutions, thus traffic analysis is the only feasible way to identify the mechanisms used to get video streams. In this context, during the 2006 FIFA World Cup, we performed an extensive measurement campaign. During this worldwide event, we measured network traffic generated by the most common P2P IPTV applications, namely PPLive, PPStream, SOPCast, an TVAnts. Our observations show that all these applications generate different traffic patterns and use different underlying mechanisms. Each application has its own download policy and maintains a different set of peers. From the traces we collected, we extract several statistics, which help in having a better understanding of the behavior of P2P IPTV systems.

1. INTRODUCTION

The success of P2P file sharing and VoIP applications has proved that the P2P paradigm is an efficient solution to deliver all kinds of content over the Internet. Nowadays, there are P2P video live streaming applications (P2P IPTV) that have been successfully deployed on the Internet. These applications are proprietary and claim to be swarming protocol like Donet [1]. In these P2P systems, data are divided into chunks and each peer exchanges with other peers information about the chunks they have. Then each peer is able to download data chunks from several peers concurrently. However, the exact way of working of these emerging applications is still widely unknown. In this context, it is important to evaluate the traffic impact of these P2P applications on the Internet. Even though lots of measurement studies have been conducted on P2P file sharing and telephony systems, very few tackled P2P IPTV.

In this paper, we make comparisons between different applications by analyzing the different traffic patterns we collected. Through these analyses, we highlight design similarities and differences and point out global behavior in these P2P networks.

During the 2006 FIFA World Cup, we measured network traffic generated by several P2P IPTV applications. We collected packet traces by using the following applications: PPLive [2], PPStream [3], SOPCast [4] and TVAnts [5]. We chose these applications because they are the most popular on the Internet. We focused on collecting traffic during World Cup because it is a large-scale event, which exhibits a live interest for the users. Thanks to all the collected data, we obtained a representative sample of large-scale P2P IPTV applications. Our work differs from Hei [6] by the number of applications we studied and the followed comparative approach. To the best of our knowledge, this is the first study, which focuses on a large-scale live event broadcasted on P2P networks.

Results in this study focus on a single event day where two soccer games were scheduled and we analyzed the traffic generated by the four previously mentioned P2P applications. We limit the scope of our analysis to these traces since we noticed their representativeness of our data set. Our results show that all the applications generate different traffic patterns. The measured download traffic indicates that the applications use different mechanisms to get the video and they maintain a different peers neighborhood.

The remainder of this paper is organized as follows. In Section 2 we present the related work. In Section 3 we describe our measurement experiment set-up. Then in Section 4 we present and discuss the measurement results. We conclude and expand our future work in section 5.

2. RELATED WORK

Measuring P2P live streaming systems is still an emerging topic, but there are previous measurement studies about live streaming media delivered on the Internet. Sripanidkulchai et al. [7] show that large-scale live streaming can be supported by P2P end-users applications despite the heterogeneous capacity of peers. In P2P IPTV systems, Zhang et al. [8] present their own P2P IPTV system and give network statistics like users' behavior in the whole system and the quality of video reception. Hei et al. [6] have a similar work to ours. They study an existing P2P IPTV application by collecting packet traces. Our work is different from theirs since we do not focus on a single application but on several applications. It helps us to highlight design differences and to infer global behavior of such P2P network without being strongly related to a single P2P system implementation. We

NOSSDAV '07 Urbana, Illinois USA

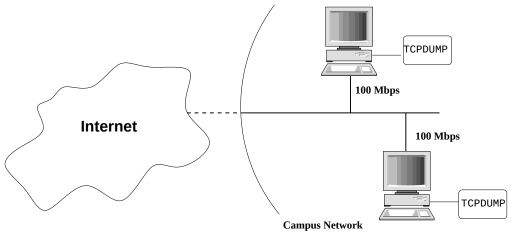


Figure 1: Measurement experiment platform. Each node is a common PC directly connected to the Internet via campus network

collected a larger and more various data set from a representative panel of applications during an entire large-scale event. Finally, an important distinction between Hei works and ours come from the live interest of the measured event. It is intuitive but corroborated by Veloso et al. [9] that traffic patterns have not the same characteristics whether broadcasted content exhibits a live interest for users or not.

3. EXPERIMENT SETUP

Our measurements started with the 2006 FIFA World Cup from June 2nd to July 9th. We collected a huge amount of data, measuring most of the World Cup games with different applications at the same time, under two different network environments (campus Ethernet access and residential ADSL access). In this paper we focus on comparisons between four P2P IPTV applications according to their traffic pattern. In all our data, we selected packet traces on June 30 because they are well representative of all of them. Two soccer games were scheduled: one in the afternoon (Germany vs. Argentine) and one in the evening (Italy vs. Ukraine). Fig. 1 describes our measurement experiment platform. We used two personal computers (PCs) with 1,8 GHz CPU, and common graphic capabilities. For the rest of this paper, the PCs will be called *nodes*. Operating system was `Windows XP` because all the applications have been implemented for this OS. The two nodes were situated in our campus network and were directly connected to the Internet with 100 Mbps `Ethernet` access. We used `tcpdump` to collect the packets and their payload generated by the applications. During each game, the nodes were running `tcpdump` and a distinct P2P application. The Ethernet cards did not suffer any packet loss and captured all the packets. All the measurements have been done watching CCTV5, a Chinese sport channel available for all the measured applications. All the applications used MPEG4 video encoding. We did not measure the traffic between the two consecutive games. From the first game to the second one, we only changed the running P2P applications on the nodes to get packet traces from different application. At the end of the experiments, we collected four packet traces: one per application. We chose to measure the first game by running PPStream and SOPCast on the nodes, and the second one by running PPLive and TVAnts. Collected packets can only provide from a node or from remote peers in the Internet. Table 1 summarizes the four collected traces.

The two measured events are similar (a soccer game in the FIFA world cup) and exhibit the same live interest for users. We analyzed our packet traces by developing our own `perl` parsing tools.

Table 1: Packet traces summary

	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	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	79.98	13.57

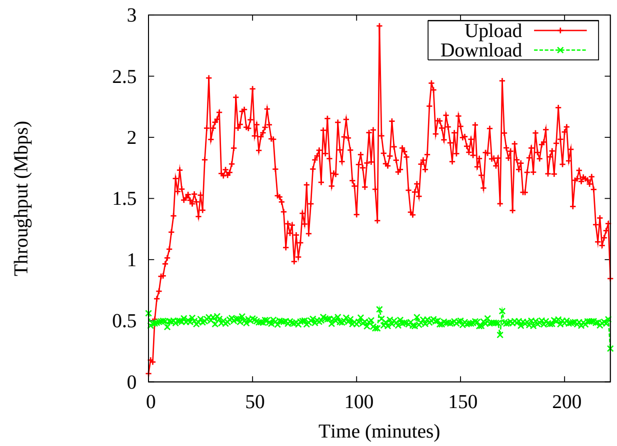


Figure 2: Example of total download and upload throughput for TVAnts. All the applications have the same throughput pattern.

3.1 Data Collection Methodology

We differentiate TCP sessions according to TCP Flags and we only take a session into consideration if at least one of its TCP segment has a payload. Session durations are driven by TCP segment payload. A session start time was calculated as soon as we received (or sent) a TCP segment with a data payload. The session duration was increased for each new TCP segment with a payload. A session ended when we received an explicit flag, but the end session time was the instant where we received the last TCP segment with payload. The session duration depends only on TCP segment with payload. We compute session duration relying on UDP in the same payload-driven way.

4. MEASUREMENT ANALYSIS

In this section, we show the results of our measurement study. We first analyze the traffic of all the applications. Then, we point out the download policies used by the applications to get the video and the peers neighborhood they maintain. Finally, we show the video peers lifetime for all the applications.

4.1 General Observations

Table 1 shows that PPStream relies only on TCP. Major part of PPLive traffic relies on TCP whereas SOPCast traffic relies mostly on UDP. TVAnts is more balanced between TCP and UDP. All the applications have download throughput quiet constant and slightly above the video bi-

Table 2: Observations Summary for Traffic Patterns

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

trate while upload fluctuates largely and at a higher rate. As an example, Fig. 2 shows the total download and upload throughput for TVAnts. The plots for all the other applications can be found in [10]. These results were expected because the nodes attempt to download video at a constant bitrate and they have wide upload capacities.

4.2 Traffic Patterns

These P2P applications are proprietary but claim to use swarming mechanisms where peers exchange information about data chunks and neighbor peers as in Donet ([1]). In such P2P network, a peer will iteratively discover other peers and would establish new signaling or video sessions. Video sessions are likely to have long duration because users want to watch the entire game whereas signaling sessions are likely to be shorter in time. Furthermore, video streaming packets size is expected to be large and signaling session packets size is supposed to be common. Fig. 3 plots the average packet size according to the session duration using a log-log scale. PPLive (Fig. 3(a)) and PPStream (Fig. 3(b)) have similar TCP traffic patterns but PPLive uses UDP too. PPLive UDP sessions vary from short to long duration, but their average packet size is small and constant. PPLive UDP traffic transports signaling sessions. PPLive and PPStream exhibit two clusters in their TCP traffic patterns: the one in the middle of the plot is for signaling sessions (small packets and short session duration) and the one in the right top of the plot is for video sessions (large packet and long session duration). PPLive and PPStream use TCP to transport video and signaling traffics. We are still investigating the difference between signaling sessions relying on UDP or TCP for PPLive. Regarding SOPCast traffic pattern (Fig. 3(c)), we observe that it uses almost exclusively UDP. We can still observe clusters in the middle (signaling) and on the right top (video) of the plot but they are not so clearly formed. SOPCast transports both signaling and video traffic on UDP. We currently have finer analysis to understand why there are very few sessions relying on TCP. Compare to the other measured applications, TVAnts traffic pattern shows a balanced use of TCP and UDP (Fig. 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 1).

Table 2 summarizes our observations from Fig 3: all the measured applications have different traffic patterns. PPStream uses only TCP for video and signaling traffic while PPLive adds UDP for some signaling traffic whereas TVAnts uses both TCP and UDP for all kind of traffic and SOPCast uses almost entirely UDP. This is an important difference between all the applications. If we take into account all the applications, we observe that the video traffic is mostly transported by TCP, which is not a transport protocol dedicated for multimedia and real-time applications. As for common Internet video streaming applications, TCP could be justified to reach all kind of Internet users, even if there are behind filtering or NAT systems.

Table 3: Signaling overhead

	PPLive	PPStream	SOPCast	TVAnts
Signaling overhead ratio	4.1%	13.6%	19.3%	10.2%

To evaluate the amount of signaling traffic overhead in the traces, we separated video and signaling traffic with an heuristic proposed in [6]. The heuristic works as follows: for a session (same IP addresses and ports), we counted the number of packet bigger or equal than 1000 Bytes[1]. If a session had at least 10 large packets, then it was labeled as a video session and we removed small packets (< 1000 Bytes) from this session. We removed all non-video sessions from the traces. Table 3 presents the results of this heuristic for the four traces. The signaling overhead ratio is different for all the traces (from 4.1% to 19.3%). SOPCast overhead is more important than the other and PPLive has the lower signaling overhead. PPStream and TVAnts have almost the same signaling overhead ratio. PPLive and PPStream have similar traffic patterns, but at the end, the signaling overhead needed to manage the P2P network is different. PPLive and PPStream should not use the same underlying mechanisms. PPStream and TVAnts have a similar overhead ratio but present different traffic patterns. Their underlying mechanisms should also be different, as SOPCast, which has the more important signaling overhead ratio and a specific traffic pattern. In the next sections, we give some insights about underlying mechanisms used by all the presented applications.

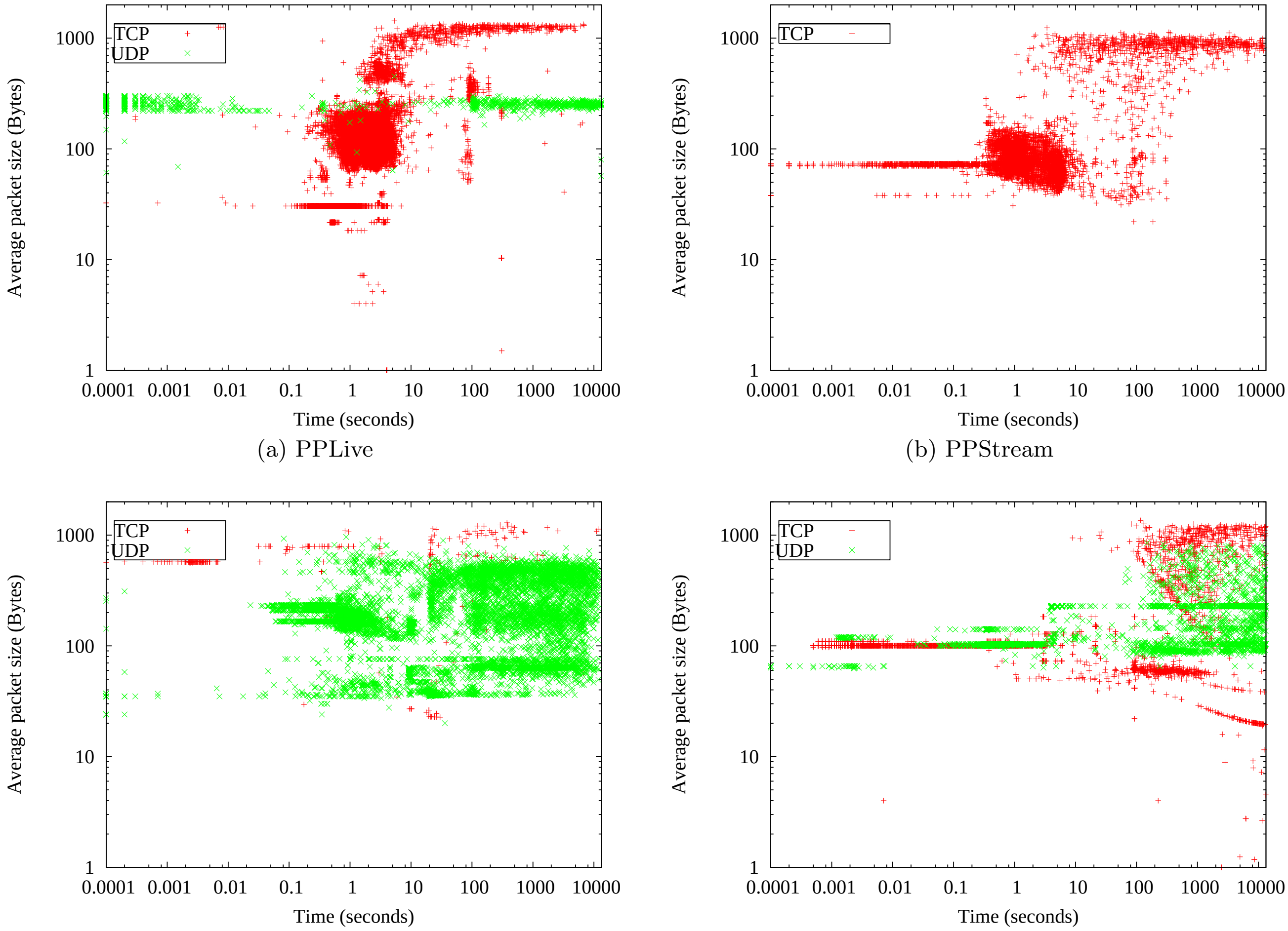
4.3 Video Download Policies

In this section, our goal is to understand how our nodes download the video among the other peers on the Internet. For each trace, we computed the amount of data that our nodes downloaded from each of the other peers. We isolated the ten top peers traffic (peers which sent the biggest amount of data to our nodes across the entire trace duration). We isolated the top peer traffic in the same way (top peer belongs to top ten peers). In Fig. 4, we plot the total download traffic, the aggregate top ten peers and the top peers download traffic. Each plotted value is a 60 seconds average interval (bin duration is 60s).

SOPCast (Fig. 4(c)) received no traffic from minutes 130 to minutes 140 and we watched a black screen during this period. The problem did not occur for network reasons because PPStream was working well during the same period. The video source has probably suffered technical problem during this period. All our SOPCast traces showed this kind of trouble and we keep them for our study.

The download policies for all the applications are totally different because aggregate top ten peers or top peer traffic do not exhibit the same behavior. For PPLive (Fig. 4(a)), the top ten peers contribute to a major part of the download

[1] 1000 Bytes instead of the 1200 Bytes proposed by the heuristic because it fitted better our traces



(a) PPLive (b) PPStream

(c) SOPCast (d) TVAnts

Figure 3: Average packet size according to peers session duration

traffic and the top peer contributes to almost all the traffic during its session duration. The top peer session is quiet short regarding all the trace duration. These observations suggest that PPLive gets the video from only a few peers at the same time and switches periodically from a peer to another one. Remember PPLive and PPStream have almost the same traffic pattern (Fig. 3(a) 3(b)), it is interesting to observe that PPStream download policy is the PPLive opposite. For PPStream (Fig. 4(b)) the top ten peers do not contribute to a large part of the download traffic and neither 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. 4(c)) contribute to about half the total download traffic and top peer contributes to all the top ten peers traffic during its session duration. In a way, SOPCast download policy looks like PPLive policy: it switches periodically from provider peer. However, SOPCast seems to always need more than a peer to get the video compare to PPLive where a single peer could be the only video provider. TVAnts download policy (Fig. 4(d)) seems to mix PPStream and SOPCast policies. TVAnts top ten peers contribute to about half the total download traffic (like SOPCast), but top peer does not contribute to a large amount of the total traffic (like PPStream). TVAnts top peer does not contribute as few as PPStream's one but does not stay as long as PPStream top peer.

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 short time (PPLive, SOPCast), or expect a peer to have huge capacities to send all the video (PPLive) or low (PPStream,TVAnts). According to the application, a peer can get the video from only a few peers at the same time or from many peers and its session duration is various. Different download policies highlight differences to maintain a neighborhood for a peer to get the video. This will be point out in the next section.

4.4 Peers Neighborhood

In swarming P2P systems, peers have to maintain peers neighborhood to get the data chunks from several peers at the same time. In Fig. 5, we plot for each application the video download peers neighborhood maintained by our nodes during all the traces duration. A video download peer is a peer, which has sent video to our controlled nodes. In

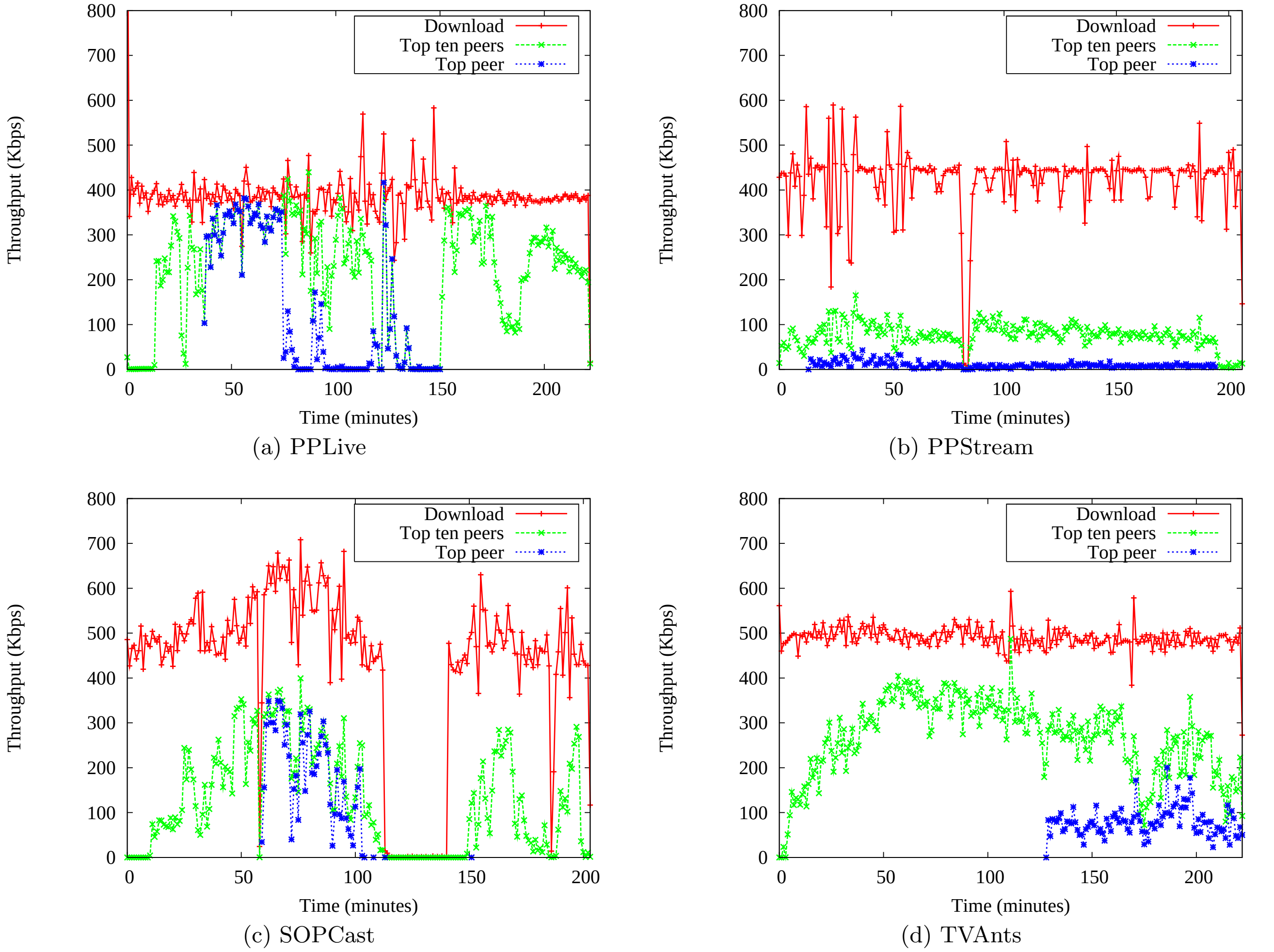


Figure 4: Video download policies: total traffic, top ten peers traffic and top peer traffic. Bin duration is 60s

the following, we will refer to the number of video download peers as *VDP*.
PPLive maintains a relatively low and constant VDP whereas PPStream has a high and constant VDP. SOPCast's VDP can be as high as PPStream's one but fluctuates largely. As expected, SOPCast has no VDP when our node running SOPCast receives no traffic. TVAnts VDP number is high and also fluctuates.
All the applications maintains for our controlled peers a different peers neighborhood, which corroborates the applications have different download policies to get the video. As expected, there is a large set of steady peers for PPStream, only a reduced set for PPLive. SOPCast and TVAnts have high and fluctuating VDP. VDP fluctuations are observed for applications, which use an important part of UDP traffic (table 1). These VDP fluctuations may come from the non reliability of UDP, which causes more packet losses and forces peer to make its VDP always evolving to get the video.

4.5 Video Peers Lifetime

In P2P IPTV, end-hosts are responsible to duplicate flows to each other. End-hosts are not entities dedicated to stay in the networks all time: they can join or leave the network whenever they want and are prone to suffer failures. P2P IPTV systems have to deal with the arrivals and departures of peers (churn of peer). It is a challenging issue because live video has to respect playback instant to get smooth playback quality. A high churn of peers will involve additional delays or jitter variations for packet delivery, which will decrease overall video quality. In this section, we show the video peers 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 time and the last time our controlled nodes exchange video traffic with another peer.
As an example, Fig. 6 plots the TVAnts complementary cumulative distribution function (CCDF) of video peers lifetime. For all the applications, the video peers lifetime CCDF follows a Weibull distribution. The CCDF plots for the other applications can be found in [10]. The Weibull distribution functions used to fit the measured video peers lifetime are presented in table 4. It also shows average peer lifetime.
The Weibull distribution is an exponential-like distribution often used in reliability testing and failure analysis. For all the applications, there are no more than 10% of peers, which stay in the network during an entire game. Moreover, the

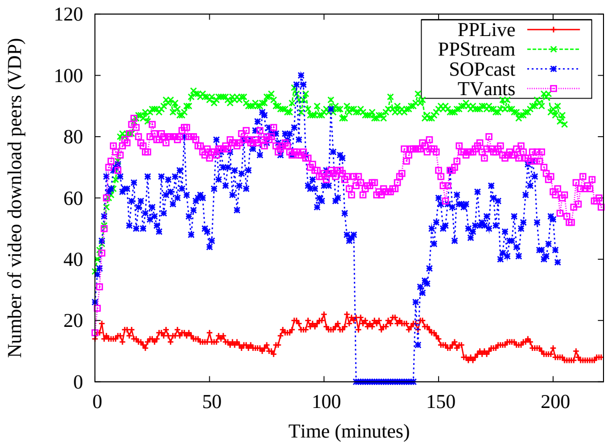


Figure 5: Peers neighborhood for all the applications. Bin duration is 60 seconds

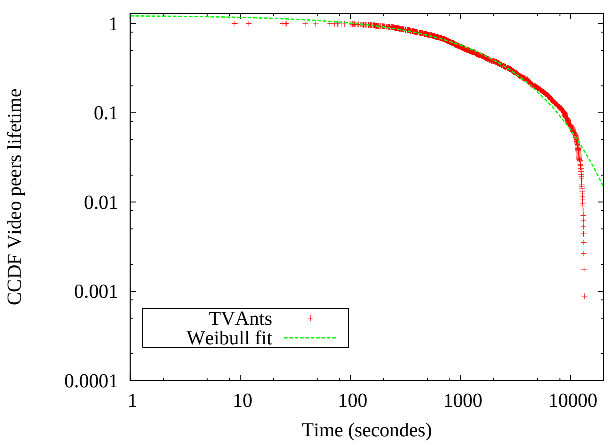


Figure 6: Example of Video peers lifetime for TVAnts. All the applications have the same Weibull-like distribution for video peers lifetime.

average video peers lifetime is different for all the applications and far from an entire game duration. The departure of a peer can be due to an user, which stops to watch the game or due to the application's mechanisms, which force a peer to switch from a video peer to another one. Since all the applications exhibit a Weibull distribution for video peers lifetime, our meaning is that Weibull distributions are driven by users' behavior of P2P IPTV applications. The mechanisms used by the applications are responsible for the different average video peers lifetime since it has been shown in this study that all the measured applications implement different mechanisms to allow peers to get the video.

5. CONCLUSION

In this paper, we explored the behavior of popular P2P IPTV systems by measuring and analyzing their network traffic. We chose the following applications: PPLive, PPStream, SOPCast and TVAnts because they are the most popular on the Internet. We measured their traffic during the 2006 FIFA World Cup since it is a large-scale event with a live interest for users. Our analyses show that the measured applications generate different traffic patterns and use different mechanisms to get the video. The application maintains different peers neighborhood and peers get the video by using different download policies. Our measurements show that for all the applications, the video peers lifetime CCDFs follow a Weibull distribution but do not have the same average time. The Weibull distribution is driven by users' behavior while the different average video peers lifetime comes from the underlying mechanisms used by the applications.
Thanks to our measurement observations, we have a better understanding of P2P IPTV systems. This knowledge will be used in our other works to model and simulate these systems.

Table 4: Video peers lifetime summary

	Video lifetime Distribution	Avg. Peer lifetime (s)
PPLive	$2.01 * e^{-(x/12.262)^{0.24}}$	393
PPStream	$1.20 * e^{-(x/322.07)^{0.39}}$	1222
SOPCast	$1.08 * e^{-(x/993.79)^{0.45}}$	1861
TVAnts	$1.23 * e^{-(x/1572.76)^{0.59}}$	2778

ACKNOWLEDGMENTS

This work is supported by the European Union under the IST Content project (FP6-2006-IST-507295).

6. REFERENCES

[1] 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. Infocom*, 2005.
[2] http://www.pplive.com.
[3] http://www.ppstream.com.
[4] http://www.sopcast.com.
[5] http://www.tvants.com.
[6] 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*, 2006.
[7] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, "The feasibility of supporting large-scale live streaming applications with dynamic application end-points," in *Proc. of SIGCOM*, 2004.
[8] X. Zhang, J. Liu, and B. Li, "On large-scale peer-to-peer live video distribution: Coolstreaming and its preliminary experimental results," in *Proc. MMSP*, 2005.
[9] 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.
[10] T. Silverston and O. Fourmaux, "P2p iptv measurement: A comparison study," http://www.arxiv.org/abs/cs.NI/0610133, 2006.