

Analysis of PPLive through active and passive measurements

Salvatore Spoto, Rossano Gaeta, Marco Grangetto, Matteo Sereno
Dipartimento di Informatica, Università di Torino
Corso Svizzera 185, 10149 Torino, Italia
email: {spoto, rossano, grangetto, matteo }@di.unito.it

Abstract

The P2P-IPTV is an emerging class of Internet applications that is becoming very popular. The growing popularity of these rather bandwidth demanding multimedia streaming applications has the potential to flood the Internet with a huge amount of traffic.

In this paper we present an investigation of the popular P2P-IPTV application PPLive exploiting a measurement strategy that combines both active and passive measures.

To this end, we use a crawler that allows the study of the topological characteristics of the overlay of one of the PPLive channels; concurrently, we perform passive measures on a PPLive client we run to join the crawled channel. We successively cross correlate information we obtained from the two measurements to assess the accuracy of the data captured by the crawler.

Our results reveal the potentials and the limits of PPLive active measures strategies.

1. Introduction

IPTV applications aim at the provision of high-quality real-time video streaming. The popularity of these applications is rapidly growing thanks to the widespread adoption of large bandwidth access links by residential customers. These applications exploit the original Internet whose infrastructure (core and access networks) could then be flooded by a huge amount of traffic.

The first attempts of deploying IPTV applications were made by ISPs exploiting IP multicast functionalities in small-scale environments through proprietary solutions based on the traditional client-server model of Internet protocols and services. The limitations of this approach are many: due to the limited deployment of IP multicast in the Internet the distribution of contents is often confined to the network of each ISP only. Furthermore, poor scalability, lack of authentication and security mechanisms, and difficult integration with hierarchical routing, makes this solution unlikely to provide the infrastructure for future massively deployed IPTV.

Recently, a different approach has emerged: IPTV applications are developed based on the popular peer-to-peer (P2P) paradigm. This model for application development requires

each participant to send data to each other using application-level multicast on overlay topologies. This approach has proved to be successful as witnessed by the steadily increasing number of applications that are nowadays available [1].

One of the most popular applications for P2P based streaming of TV programs is PPLive [2]. PPLive is a mesh-pull P2P streaming platform that distributes both live and pre-recorded contents. According to the PPLive web site [2] in January 2008 the PPLive application provided almost 500 channels with 1,000,000 daily users on average. The number of channels in December 2008 was reported to be equal to 1775. The bit rates of video programs mainly range from 250 kbps to 400 kbps with a few channels as high as 800 kbps.

The above data about PPLive make it clear that it is important to analyze how these applications organize to be able to evaluate their impact on the traffic loading the Internet infrastructure. A methodology for the investigation of an existing IPTV system is measurement of relevant parameters that could be exploited to build simulators and/or analytical models of P2P based IPTV applications to support the design and implementation of more efficient systems. In this paper we present a measurement study of PPLive that combines the use of passive and active measures to derive information on the traffic as well as on the overlay topology generated by PPLive channels. To this end we develop a distributed analysis software that collects information on the topological properties of PPLive overlay networks, i.e., a crawler of PPLive overlays. This active measurement experiment is correlated with the concurrent measures obtained by monitoring a peer joining the crawled overlay (passive measurement). The combined use of passive and active measures allows us to discuss the limits and the accuracy of the information that can be collected by using active measures for PPLive and this provides a better interpretation of results presented in previous studies. We report findings that confirm results obtained by other investigators as well as a characterization of the user behavior in the time domain.

The outline of the paper is the following: in Section 2, we briefly summarize previous findings from measurement-based investigations of P2P-IPTV applications. Section 3 provides a short description the PPLive protocols and of the software we used to capture snapshots of the PPLive overlay. Section 4 describes the measurement strategy that

combines both active and passive measures and the insight we derived by using this strategy. Finally, Section 5 synthesizes connections throughout our work and outlines several developments for current and future investigation.

2. Related Works

P2P file sharing overlays have been the subject of numerous investigations that use active measurement strategies. In this field the application that received most of the attentions has been the Gnutella file sharing overlay. There are two main motivations for this: the popularity and the availability of the protocol details. Some results in this field provide interesting lessons that can be also used for studies of other P2P overlay. In particular, the work presented in [3] points out that the crawling speed may affect the accuracy of the collected results. The distributed architecture of the crawler that we use for crawling the PPLive overlay is inspired by the one proposed in [3] to explore the Gnutella overlay topology.

Since virtually every existing P2P-IPTV application is proprietary, active measurements of P2P-IPTV systems are very difficult due to the unavailability of protocol details. In fact, numerous papers have been published presenting passive measurements of P2P-IPTV systems, e.g., [4], [5] and references therein. On the other hand, very few previous works have been developed on the subject of active measurements of P2P-IPTV systems. Papers [6] and [7] are remarkable exceptions. The work in [6] mainly focuses on traffic characteristics of PPLive peers in a time limited window. Results are presented for a very popular TV program retransmitted on a PPLive channel. The crawler details are not presented for comparison. The work presented in [7] defines a crawling strategy for PPLive channels based on reverse engineering of application level data packet to extract relevant protocol features. This work represent the starting point for our investigation. In fact, we modified their crawler (that is available for download at <http://cairo.cs.uiuc.edu/~longvu2/pplive.html>) according to the lessons we learned from [3]. Furthermore, this paper presents results on the topology of overlay networks representing sample TV channels and a characterization of the user behavior.

The main contribution of our paper is the refinement of the crawling strategy developed in [7] to confirm results presented therein as well as to discuss on the accuracy of active measurements aimed at the characterization of overlay network topology.

3. PPLive Description and Measurement Methodology

A common approach for measurement based investigation of P2P systems is to capture snapshots of the overlay network using a crawler, i.e., a program that collects information on the topological properties of the overlay network.

The development of a crawler requires the knowledge of protocols used by the application under study. In the PPLive case this is a challenging task because this application is not open source and its details are not public. In the following we first provide a short description of what is known about the PPLive protocols (Section 3.1) and then we describe the design of our crawler (Section 3.2).

3.1. Short Overview of PPLive

PPLive is a mesh-pull P2P streaming platform that distributes both live and pre-recorded contents. According to the PPLive web site [2] in January 2008 the PPLive application provided almost 500 channels with 1,000,000 daily users on average. The number of channels in December 2008 was reported to be equal to 1775. The bit rates of video programs mainly range from 250 kbps to 400 kbps with a few channels as high as 800 kbps. The PPLive is a proprietary system, which means that neither the source code nor protocols are available. The following description of the behavior of PPLive is derived by our experiments and by previous investigation that use passive and active measurements (see for instance [6], [4], and [7]).

The PPLive platform consists of multiple overlays. A single overlay corresponds to a PPLive channel. Each peer in an overlay is identified by the pair (IP address, port number). Figure 1 shows the basic actions of a PPLive peer: (1) channel list download from the *channel list server* (via http); (2) for the selected overlay (channel) the peer collects a small set of peers involved in the same overlay by querying the *membership servers* (via UDP); (3) the peer communicates with the peers in the list to obtain additional lists which it aggregates with its existing peer list (via UDP). In this manner the peer maintains a list of other peers involved in the same overlay (i.e., watching the same channel). During the streaming the peer periodically executes steps (2) and (3) to collect neighbors peers candidate.

3.2. The PPLive crawler

As discussed PPLive is a closed source application whose details are unknown. To overcome this difficult we use the technique presented in [7] that allows to collect the query strings that the peers use in steps (2) and (3)¹. Unfortunately, this is not the only problem to solve to obtain accurate data from the crawling experiments. In fact, according to the studies of the Gnutella P2P file sharing application presented in [3] and [8], the accuracy of captured overlay snapshots depends on the crawling speed. In principle, a perfect snapshot is captured if the crawling process is instantaneous and

1. Unfortunately, the number of PPLive channels that can be investigated using the method described in [7] is decreasing, this is probably due to changes in internal protocols used by PPLive.

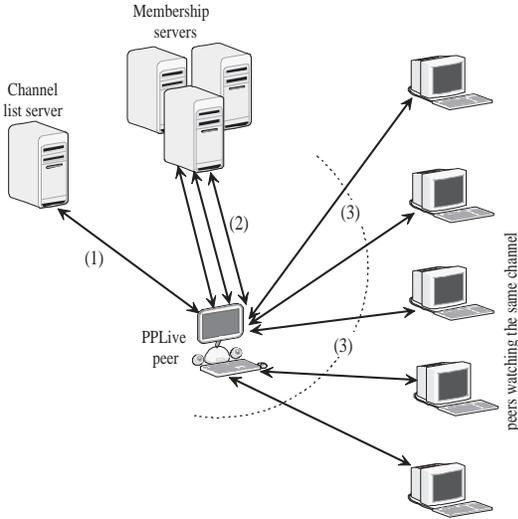


Figure 1. PPLive basic architecture

complete, i.e., if all peers are contacted. However, neither of these conditions is generally met for the following reasons: rapidly changing topology, and unreachable peers. To obtain accurate overlay snapshots, we used a distributed crawler architecture, similar to the one described in [3]. The PPLive crawler employs a master-slave architecture to achieve a high degree of concurrency. For our experiments, we used 5 PCs that run 1 master and 200 slaves. The master process performs step (2) of Figure 1 to collect the initial list of peers and then coordinates multiple slave processes. In particular, each slave receives peer addresses from the master according to the slave crawling speed and performs the step (3) of Figure 1. The slave processes are independent crawlers that probe different portions of the overlay. Moreover, to increase the crawling speed each slave process is composed by a set of threads that independently send queries to PPLive peers. Periodically, each slave process reports back to the master process with the peer neighbors.

Unresponsive peers make the snapshots less accurate. Peers may not respond to queries from our crawler for several reasons. If a peer is temporarily overloaded it might later respond if we adopt a high timeout value. On the other hand, a too high timeout may introduce long delays. We performed a set of preliminary experiments to derive a timeout value to trade-off the increase of the number of responding peers and the decrease of long delays. In our experiments we set the timeout equal to 5 sec.

A related issue concerns the difficulty in obtaining the neighbor list. In particular, as observed in [7] and confirmed in our experiments, successive queries sent to the same peer may yield slightly different list of neighbors. We performed a set of experiments to derive a value for the number of successive queries that represents a trade-off between the

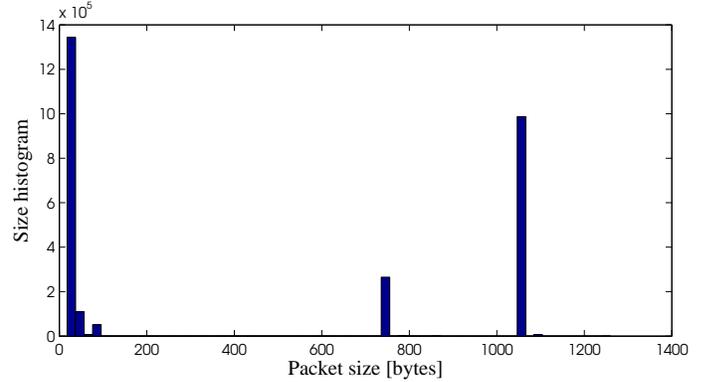


Figure 2. PPLive packet size histogram

need to obtain the peer’s neighbor list and the necessity of having short crawling time. In our experiments we set the number of successive queries sent to the same peer equal to 10. Furthermore, we address in more detail the “completeness” of the the peer’s neighbor lists in Section 4. With all the optimizations included in our crawler we observe that the average duration of each experiment is 106 sec, with an almost negligible standard deviation.

4. Crawling “completeness” and passive/active measures

In this section we provide results on the amount and type of data exchanged by a controlled PPLive client whose activity is traced by a sniffing packet tool. The data collected during these passive measurements are exploited to define the concept of neighbor and to compute the intersection between data obtained by the active crawling and passive monitoring. This latter task permits to evaluate the completeness of the active measurements. Furthermore, active measurements allows us to report on the topological properties of sample overlay networks and on the time behavior of peers joining these channels.

4.1. Passive measurements

For the passive measurements we use a PPLive running on a non-natted LAN node. We only capture UDP traffic on the known PPLive port number. In the first experiment we capture the in/out traffic of the PPLive node watching a cartoon channel. In Figure 2 the packet size histogram of the captured packets during a session of 1500 s is shown. Three packet types can be identified: < 100 bytes (signaling, class A), around 800 bytes (to be identified, class B), > 1000 (video data, class C). During the same session we observe an overall download rate of 616 kbps whereas the upload rate reaches 6082 kbps. Table 1 shows the partial rates obtained considering the three packet classes identified in Figure 2. It

Table 1. Rates [kbps] for the packet class A,B,C.

Packet class	A	B	C
uplink	42.79	964.98	5116.81
downlink	196.51	64.76	355.07

is very likely that class C represents video data encoded at 350 kbps. Class A is probably to be referred to the signalling for the overlay network management and control (neighbor lists); it is worth pointing out that such control traffic is larger in the downlink than in the uplink. Class B deserves further attention; it can be noticed that the class B traffic represents a constant fraction of the class C rates, i.e. video packets. In particular, class B amounts to the 18% of the video traffic in both the downlink and uplink. This allows us to infer that class B carries data which are strictly correlated to the video; these are likely to be the buffer maps, which are exchanged among peers in a fixed proportion with respect to the video rate.

In Figure 3 and Figure 4 the rate from and toward the different peers is shown; each peer is mapped onto a unique positive integer identifier reported on the x-axis; on y-axis positive are used for the uplink, whereas negatives refer to the downlink rates, measured in kbps. It can be noticed that class A rates are far more uniform with respect to class B, C. This is due to the fact that control packets evenly spread around the overlay. On the contrary, video and buffer maps traffic is more peaked around few peers which the probed node is collaborating with. In the observed time window about 2000 peers are contacted by our node. Nevertheless, the number of active neighbors is rather limited; as an example, by considering only those peers exhibiting class B/C rates larger than 1 kbps, one retains only 15% of the observed 2000 peers. This characteristic is confirmed in Figure 5, which shows the rate generated by class C packets for each peer. In Figure 5 the uplink and downlink rates per peer are sorted in decreasing order, pointing out that for the node we monitor a large fraction of data is exchanged with only a small subset of its neighborhood.

4.2. Combination of active and passive measures

The goal of crawling experiments is to obtain a snapshot of the overlay topology that could also be used to derive a graph representation of the overlay network. To this end, peers are queried to provide the list of their neighbors. Who are these neighbors? To answer we analyzed the trace obtained from the passive measurements described in Section 4.1 and we computed the number of neighbors by means of a proper definition of the neighbor concept. The neighbor is defined as follows.

Definition 1. A peer u is a neighbor of our controlled PPlive client if and only if in a time window starting at time t

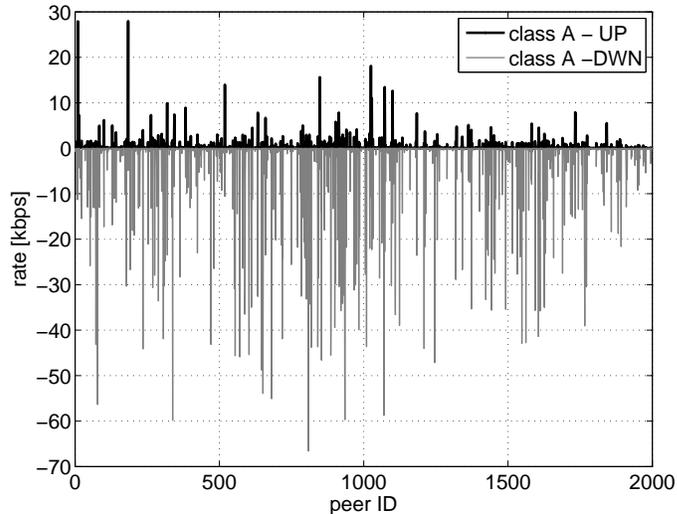


Figure 3. Class A rates [kbps] per peer.

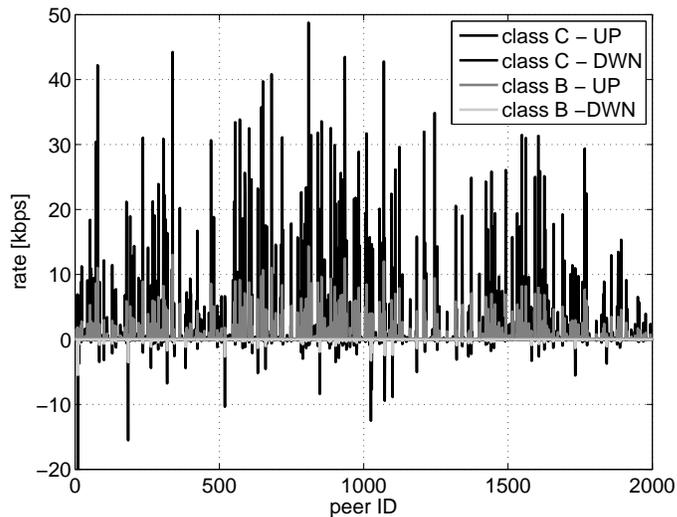


Figure 4. Class B-C rates [kbps] per peer.

and lasting for Δ seconds at least one packet exchange is reported.

Clearly, this definitions yields different concepts of neighbor depending on the direction of the video packet exchange and on the class of the packet being exchanged (class A, B or C). Furthermore, this definition is dependent on both t and Δ . In Figure 6 the number of neighbors is reported versus time for $\Delta = 15$ seconds by considering only class C packets exchange and for both directions of packet exchange. Furthermore, the number of peers that are contacted for both upload and download are represented by the intersect curve. It can be observed that after an initial transient time interval, where the cardinality of the neighborhood is rapidly growing, the measures reach a steady state where slight

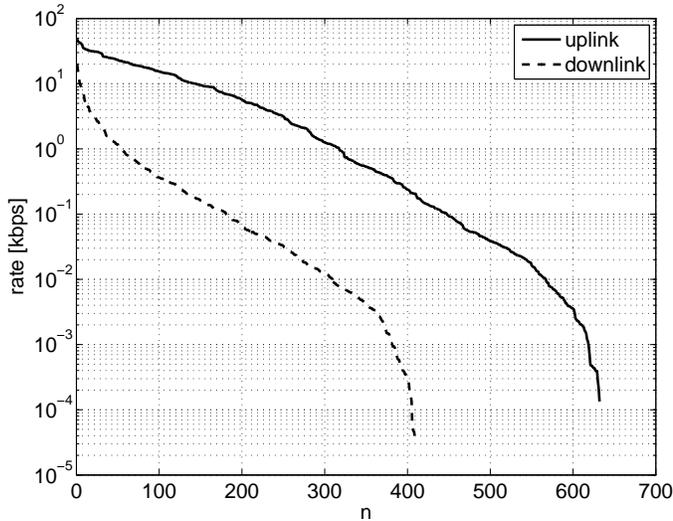


Figure 5. Class C sorted rates [kbps] per peer.

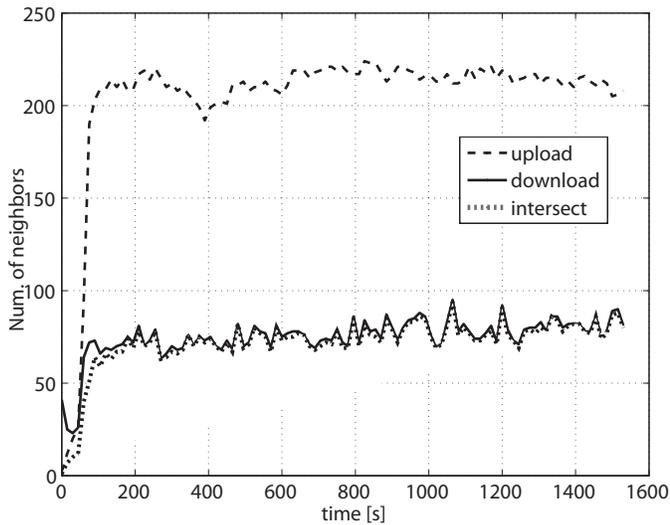


Figure 6. Number of neighbors versus time for $\Delta = 15$ seconds.

oscillations around an equilibrium value are noted.

In order to compare active and passive measurement results we performed the following experiment. With the same settings described in Section 4.1 we started a crawling of the cartoon channel overlay network after a time interval that allowed our controlled PPLive client to reach the steady state. At the end of the crawling we computed the intersection between the list of neighbors obtained from the crawler for our controlled client and the list of neighbors we compute from the passive measurement according to definition 1. We consider different values for Δ (10, 20 and 30 seconds) and throttle download and upload rates of our PPLive client by means of the tool *NetLimiter* [9].

Table 2 shows the number of the neighbors of the controlled node (the node degree) obtained by the crawler. The same table reports the intersection in percentage between the set of neighbors obtained by active and passive measurements. It can be noted that:

- the lower the bandwidth, the lower the node degree is;
- the largest overlapping between the passive and active measures occurs for packets belonging to class C;
- the intersection percentage is almost insensitive to the values of Δ we considered;
- a measurement strategy for PPLive that uses a crawler yields only partial information about the actual overlay links. Therefore active measurements must be used with care in order to infer overlay features e.g., topological characteristics.

4.3. Active measures and topological properties

As we observed in the previous section the accuracy of the crawling experiments can be rather poor with neighbors intersection that can be as low as about 40%. On the other hand the accuracy of the experiments also depends on the percentage of non-responding peers, that is peers whose identity has been discovered during the exploration of the overlay network (i.e., from another peer or from the membership servers) but that deny (or excessively delay) their reply to a request from the crawler. In the following we present results obtained from a group of successive measurements of the same PPLive overlay channel (in particular we focus the attention on a cartoon channel) acquired at a constant rate. The duration of this battery of snapshots derives from the crawling time of each snapshot and from the constant time interval between successive acquisitions. We identify each snapshot with a progressive number (ID). These crawling experiments were collected between 11/02/2008 and 11/11/2008; during this period we collected 7670 snapshots of the PPLive overlay channel under study. The average duration of the crawling experiments was 106 sec (with a small standard deviation). The plots of Figure 7 show a typical daily behavior.

From these experiments we found that the fraction of responding peers varies from the 83% to the 52%. We cannot discriminate excessive long delay from unreachable peers but with a time out chosen to minimize these misclassifications we found the fraction of unresponsive peers is mostly due to unreachable peers (probably NAT-ed peers).

In Figure 8 we plot the degree distribution computed from the collected snapshots. In particular, we consider two PPLive channels, one the cartoon channel that we previously described (channel 1), and a news channel (channel 2).

To be representative of the "real" overlay PPLive topology the results depicted in the previous figure should be "corrected" according to the considerations summarized in Table 2.

Table 2. Average node degree of the monitored node and intersection (percentage) between the set of neighbors obtained via active and passive measurements.

Download/Upload bandwidth, Δs	Node degree	Neighbors intersection		
		class A	class B	class C
1Mb/0.25Mb, 30s	93	7%	50%	78%
1Mb/0.25Mb, 20s		9%	50%	74%
1Mb/0.25Mb, 10s		13%	50%	73%
1Mb/0.5Mb, 30s	94	15%	33%	88%
1Mb/0.5Mb, 20s		23%	33%	91%
1Mb/0.5Mb, 10s		39%	33%	90%
1Mb/1Mb, 30s	102	5%	< 1%	35%
1Mb/1Mb, 20s		7%	< 1%	36%
1Mb/1Mb, 10s		9%	< 1%	37%
4Mb/0.25Mb, 30s	102	12%	25%	61%
4Mb/0.25Mb, 20s		13%	25%	61%
4Mb/0.25Mb, 10s		20%	33%	61%
4Mb/0.5Mb, 30s	115	11%	10%	59%
4Mb/0.5Mb, 20s		15%	34%	60%
4Mb/0.5Mb, 10s		19%	< 1%	63%
4Mb/1Mb, 30s	115	11%	50%	55%
4Mb/1Mb, 20s		13%	50%	54%
4Mb/1Mb, 10s		18%	50%	54%
Lan, 30s	121	10%	40%	40%
Lan, 20s		16%	40%	44%
Lan, 10s		12%	40%	44%

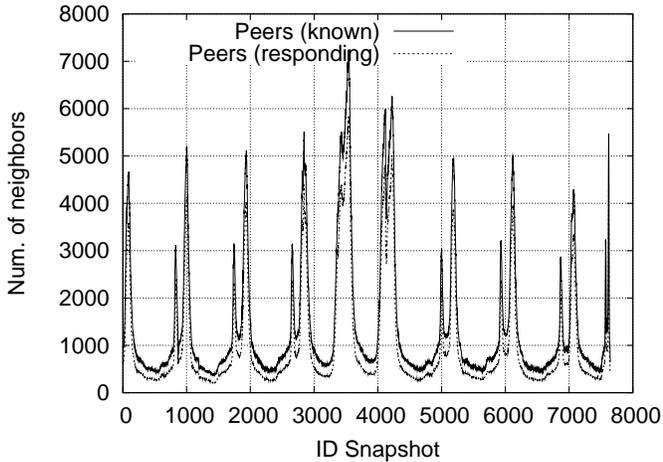


Figure 7. Evolution of peer population over nine days

We also studied the length of users connection to the PPLive overlay. A session is characterized by the presence of the same node, identified by its IP address and UDP port number, in a set of consecutive snapshots. Every snapshot has an average duration of 106 sec, in which a peer is considered active and connected. The lengths of connected sessions have been already studied by using a similar approach in [7]. We obtain results similar to those presented in that paper, i.e., the duration of sessions on a PPLive channel (we perform the same experiment for different channels)

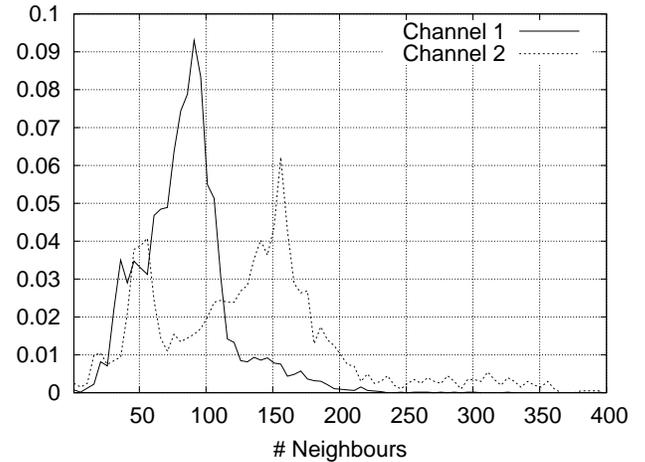


Figure 8. Peer degree distribution (for two PPLive channels)

can be approximate by means of a exponential distribution (or geometric as reported in [7]). In particular we plot the session length distribution computed on the battery of 7670 snapshots in Figure 9.

5. Conclusions and further developments

PPLive is a P2P-IPTV platform that provides users with hundreds of TV channels. Each TV channel is supported by an overlay network composed of peers watching the channel

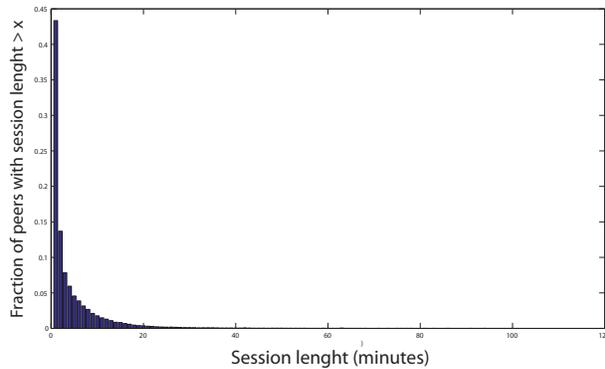


Figure 9. Peer session duration

programs. These applications are likely to become one of the most important traffic sources for the Internet. In this paper, we presented a passive/active measurement study regarding the characteristics of the PPLive overlay networks. To this end, we used a distributed crawler to capture snapshots of PPLive overlay networks supporting TV channels. The crawler is based on previous work on this subject and has been refined to account for the techniques developed in the context of crawling P2P file sharing applications.

As a sort of validation we re-derived results on the length of users connection to PPLive channels: the preliminary results we obtained confirm those in [7], i.e., the session lengths can be represented by means of memoryless distributions (geometric or negative exponential). The main contribution of the analysis presented in this paper is the study of the limits and the completeness of active measurements acquired crawling the PPLive overlays. The presented results showed that a simple crawling strategy is not able to get accurate snapshots of the PPLive. The lesson that we can draw is that the derivation of accurate topological information for PPLive (and possibly for similar P2P-IPTV applications) requires the setting of a more sophisticated measure methodology. This is the main direction of our future investigations.

References

- [1] “PPStream, <http://www.PPStream.com>; PPLive, <http://www.pplive.com>; SOPCast, <http://www.sopcast.com>; TVAnts, <http://www.tvants.com>; Joost, <http://www.joost.com>; Babelgum, <http://www.babelgum.com>; Zattoo, <http://www.zattoo.com>; Tvunetworks, <http://www.tvunetworks.com>.”
- [2] “PPLive internet site,” Registered in 2004, <http://www.pplive.com/en/>.
- [3] D. Stutzbach and R. Rejaie, “Capturing Accurate Snapshots of the Gnutella Networks,” in *Proc. of Global Internet Symposium*, 2005.
- [4] D. Ciullo, M. Mellia, M. Meo, and E. Leonardi, “Understanding P2P-TV Systems Through Real Measurements,” in *Proc. GLOBECOM 2008*, New Orleans, FL, USA, Nov 2008.
- [5] C. Wu, B. Li, and S. Zhao, “Characterizing peer-to-peer streaming flows,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, 2007.
- [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,” *IEEE Transactions on Multimedia*, vol. 9, no. 8, 2007.
- [7] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt, “Measurement and Modeling a Large-scale Overlay for Multimedia Streaming,” in *Proc. of the Int. Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine 2007)*, Vancouver, BC, Canada, Nov 2007, details available at <http://cairo.cs.uiuc.edu/~longvu2/pplive.html>.
- [8] D. Stutzbach, R. Rejaie, and S. Sen, “Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems,” in *Proc. of the ACM SIGCOMM Internet Measurement Conference*, Berkeley, CA, USA, 2005.
- [9] “NetLimiter Bandwidth Shaper,” Shareware edition, <http://www.netlimiter.com/>.