Understanding Overlay Characteristics of a Large-scale Peer-to-Peer IPTV System

Long Vu[†], Indranil Gupta[†], Klara Nahrstedt[†], Jin Liang[‡]
[†]Department of Computer Science, University of Illinois
[‡]Google Inc.

This paper presents results from our measurement and modeling efforts on the large-scale peer-to-peer (p2p) overlay graphs spanned by the PPLive system, the most popular and largest p2p IPTV (Internet Protocol Television) system today. Unlike other previous studies on PPLive, which focused on either network-centric or user-centric measurements of the system, our study is unique in (a) focusing on PPLive overlay-specific characteristics, and (b) being the first to derive mathematical models for its distributions of node degree, session length, and peer participation in simultaneous overlays.

Our studies reveal characteristics of multimedia streaming p2p overlays that are markedly different from existing file-sharing p2p overlays. Specifically, we find that: (1) PPLive overlays are similar to random graphs in structure and thus more robust and resilient to the massive failure of nodes, (2) Average degree of a peer in the overlay is independent of the channel population size and the node degree distribution can be fitted by a piecewise function, (3) The availability correlation between PPLive peer pairs is bimodal, i.e., some pairs have highly correlated availability, while others have no correlation, (4) Unlike p2p file-sharing peers, PPLive peers are impatient and session lengths (discretized, per channel) are typically geometrically distributed, (5) Channel population size is time-sensitive, self-repeated, event-dependent, and varies more than in p2p file-sharing networks, (6) Peering relationships are slightly locality-aware, and (7) Peer participation in simultaneous overlays follows a Zipf distribution. We believe that our findings can be used to understand current large-scale p2p streaming systems for future planning of resource usage, and to provide useful and practical hints for future design of large-scale p2p streaming systems.

Categories and Subject Descriptors: C.2.4 [**Computer Systems Organization**]: Computer Communication Networks—*Distributed Systems*

General Terms: Measurement, Performance

Additional Key Words and Phrases: Peer-to-Peer, IPTV, Streaming, Multimedia, Overlay, PPLive

1. INTRODUCTION

The proliferation of large-scale peer-to-peer (p2p) overlays such as Kazaa, Gnutella, Skype, PPLive [PPL], PPStream, TVUPlayer, Sopcast, CoolStream, and RONs [Andersen et al. 2001] has created the need to characterize and understand the emergent properties of these overlays. A large fraction of existing characteristic studies focus on file-sharing p2p applications, such as Kazaa, Gnutella, and Napster.

Authors' addresses:
- Long Vu, Indranil Gupta, Klara Nahrstedt, Department of Computer Science, University of Illinois, 201 N. Goodwin Avenue, Urbana, IL 61801, USA. Email: {longvu2,indy,klara}@illinois.edu
- Jin Liang, Google Inc., Email: jinliang@gmail.com

Some of the more prominent studies among these are by Ripeanu et. al. [Ripeanu et al. 2002] on Gnutella, by Saroui et. al. on Naspter and Gnutella [Saroiu et al. 2003], and by Bhagwan et. al. on Overnet [Bhagwan et al. 2003]. Although these studies have created a better understanding of the characteristics of p2p overlays, there is a risk that some system designers may believe that the conclusions drawn from above studies are shared by many other p2p overlays such as p2p streaming overlays.

This paper shows that many of the well-held beliefs about the characteristics of p2p file-sharing overlays may be false when one changes the application atop the p2p streaming overlays. Specifically, we undertake a crawler-based study of a deployed application overlay network for IPTV, called PPLive. We believe that results obtained from our studies can be used to understand large-scale p2p streaming systems for future planning of resource usage, and to provide useful and practical hints for future design of large-scale p2p streaming systems.

P2P IPTV applications have seen a dramatic rise in popularity and have received significant attention from both industry and academia. The number of subscribers is predicted to increase from 3.7 million in 2005 to 36.9 million by 2009 [Mul]. This promising market has encouraged the rapid development of IPTV technologies including tree-based multicast [Banerjee et al. 2002; Tran et al. 2003], receiver-driven p2p streaming [Liang and Nahrstedt 2006; Rejaie and Stafford 2004] and chunk-driven p2p streaming [Zhang et al. 2005; Li et al. 2008]. Among these, the chunk-driven approach has emerged as the most successful technology with a large number of simultaneous viewers [Hei et al. 2007].

PPLive is a chunk-driven p2p IPTV streaming system, which stands out due to the heterogeneous channels and increasing popularity. As of May 2006, PPLive had over 200 distinct online channels, a daily average of 400,000 aggregated users, and most of its channels had several thousands of users at their peaks [PPL]. During the Chinese New Year 2006 event, a particular PPLive channel had over 200,000 simultaneous viewers [Hei et al. 2007]. In our experiments from February 2006 to May 2008, we observed that there were between 400 and 500 daily online channels, with 400,000 to 500,000 aggregated simultaneous viewers.

There have been several measurement studies done on the PPLive streaming system [Hei et al. 2007; Ali et al. 2006; Silverston and Fourmaux 2007; Huang et al. 2008], which tend to predominantly look at either network-centric metrics (e.g., video traffic, TCP connections, etc.), or at user-centric metrics (e.g., geographic distribution, user arrival and departure, user-perceived quality, etc.). Our crawler-based measurement studies therefore are unique in focusing primarily on *overlay-based characteristics* of the PPLive streaming system, which is related to, yet different from, the user-centric view and the network centric view. Of course, overlay characteristics are influenced by an amalgamation of both user behavior and by the design of the underlying protocol and the network, yet they stand apart themselves. Our studies also expose new avenues for improving performance, reliability, and quality of IPTV systems in the future. Moreover, to the best of our knowledge, we are the first to provide mathematical models for the overlay characteristics of p2p IPTV systems.

Results obtained from our extensive experiments (stretching from February 2006 until May 2008) indicate that PPLive overlay characteristics differ from those of p2p file-sharing. Our major findings are: (1) PPLive overlays are similar to random graphs in structure and thus more robust and resilient to the massive failure of nodes, (2) Average degree of a peer in the overlay is independent of the channel population size and the node degree distribution can be fitted by a piecewise function, (3) The availability correlation between PPLive peer pairs is bimodal, i.e., some pairs have highly correlated availability, while others have no correlation, (4) Unlike p2p file-sharing peers, PPLive peers are impatient and session lengths (discretized, per channel) are typically geometrically distributed, (5) Channel population size is time-sensitive, self-repeated, event-dependent, and varies more than in p2p file-sharing networks, (6) Peering relationships are slightly locality-aware, and (7) Peer participation in simultaneous overlays follows a Zipf distribution. All the above conclusions, except (2), are markedly different from the well-known characteristics of p2p file-sharing systems.

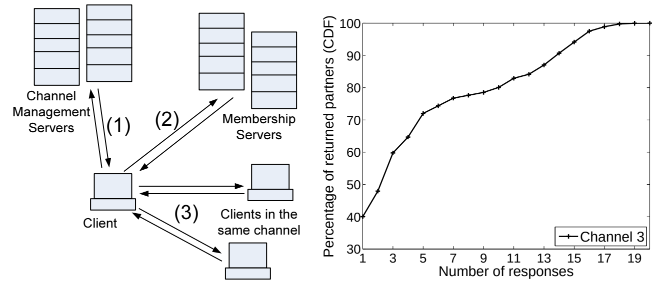
In this paper, we first describe PPLive basics and preliminary definitions (Section 2) and present our measurement methodology (Section 3). Then, we study the characteristics of the PPLive overlay at three different levels: that of an individual node, that of node pairs, and that of the entire overlay. Particularly, we study node level overlay characteristics by presenting and modeling the node degree distribution, overlay randomness, and node session length (Section 4). We study the overlay characteristics of node pairs by investigating peer availability interdependence and locality-awareness of PPLive peers in choosing streaming partners (Section 5). Next, we study the overlay characteristics from system-wide level in Section 6. Specifically, we study the variation of the channel population size over time, distributions of the peer participation in simultaneous overlays, and the resilience of PPLive overlays under the massive failure of nodes. After that, we present the related work in Section 7. Finally, we conclude and draw lessons for future design of p2p streaming systems in Section 8.

2. PPLIVE BASICS AND PRELIMINARY DEFINITIONS

Before embarking on our study of PPLive, we briefly summarize its basic architecture as well as the structure of its content channels. In each case, we provide basic definitions that will be reused later in the paper.

2.1 PPLive Overview

PPLive is a free, close source p2p IPTV application, which divides video streams into chunks and distributes them via overlays of cooperative peers. The PPLive system consists of multiple overlays, in which each content channel is associated with one overlay. Each channel streams either live content or a repeating prefixed program, and the feed from the channel may originate from one or multiple sources. Similar to TV users, a PPLive user can join at most one channel at one time. This viewing behavior differs from other multimedia systems where a user can view simultaneous channels in multiple windows. In our experiments (Section 6.2), we observe that when a PPLive user watches a channel, her client machine is not only a consumer of feeds from that channel, but may also be chosen by the protocol to act as a relay for feeds from other channels. That is, the per-channel overlay might include its own subscribers and a few others, which do not subscribe to that overlay.



(a) PPLive protocol (b) k response degree

Fig. 1. PPLive protocol and k response degree

By default, each PPLive client has a pair of TCP and UDP ports (per channel) to communicate with PPLive servers and its neighboring peers. A number of other TCP ports can be used by the client to exchange video chunks during its sessions.

There are several challenges in studying PPLive overlays. Particularly, it is very difficult to distinguish between the notion of a "user" and a "client machine". There are several reasons for this: (1) PPLive users are free to join, leave, and switch channels by accessing the PPLive web interface or PPLive Net TV player. (2) Due to NATs and firewalls, a user's client machine may change its IP or UDP port number or both. (3) The proprietary PPLive system is widely believed to use the idea of inter-overlay optimizations (we also observe this in our experiment in Section 6.2) in order to recruit non-subscribing nodes, which is used by the Anysee system [Liao et al. 2006]. As a result, a client machine may appear as a participant in multiple overlays, including ones that the user is not subscribed to. Hence, in the rest of this paper, we refer to a given $< IP, port >$ tuple as a "node" or a "peer" - this is a combination of both a client machine and a user. The term "client" refers only to the machine (e.g., workstation) that the PPLive player is running on, while "user" refers to the human user - these should not be confused with node or peer.

2.2 PPLive Membership and Partnership Protocols

In the streaming system, each PPLive peer executes two protocols, for (1) registration and harvesting of partners, and (2) p2p video distribution. For our studies, we develop a crawler, which follows the first protocol to crawl peers attending PPLive content channels. Before discussing the first protocol in details, we define the notion of a partner of a peer as follows. A peer p_2 is considered a partner of a peer p_1 if (1) p_2 uploads streams to p_1 or p_2 downloads streams from p_1 or both, in this case p_2 is denoted a streaming partner, or (2) p_2 exchanges the control data (e.g., signaling traffic) with p_1, in this case p_2 is denoted a signaling partner. In our study, we leverage a PPLive API, which allows a peer to be queried for its partner list. The partner list of a peer p_1 is defined as a list of partners returned by p_1 when it gets queried for the partner list. The above definition of partners results from two facts: (1) since PPLive is close in nature, how a peer manages and returns its partners when being queried is unknown, (2) from the format of the UDP packets [Ali et al. 2006] returned by a PPLive peer p_1 when p_1 gets queried for its partner list, it is impossible to differentiate between streaming and signaling partners of p_1.

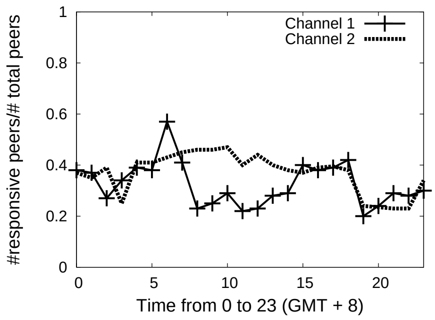
Figure 1(a) shows the first protocol (registration and harvesting of partners) executed at a client p in the PPLive network: (1) p retrieves a list of channels from channel management servers via $HTTP$, (2) for its interested overlay, p retrieves a set of nodes from the membership servers via UDP, (3) p uses this seed partner list to harvest (learn about) other partners in the same channel by periodically probing existing partners via UDP. During its streaming session, p may also sometimes perform step (2) and step (3) simultaneously to obtain potential partners from membership servers and existing partners. If a PPLive node is inside a NAT or a firewall, UDP in the above steps may be replaced by TCP.

2.3 PPLive Overlay

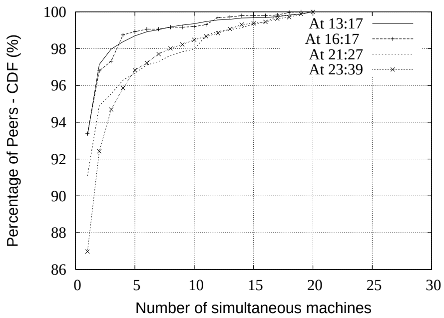
We assume that a PPLive overlay for a content channel is a directed graph $G = (V, E)$. Recall that each PPLive overlay corresponds to an individual PPLive channel. Here V is the set of nodes attending the overlay and E is the set of links between nodes. Each node (or peer) is defined as a given $< IP, port >$ tuple and belongs to V. Each partner of a node p, appearing in p's partner list, then corresponds to an edge (or link) in E. This directed graph G is used instead of an undirected graph since: (1) the knowledge of PPLive overlay formation protocols is proprietary, therefore, we opt for a more general and safe possibility (e.g., the directed graph), (2) the directed graph is also a natural consequence of the membership protocols (at minimal, a peer p_1 might obtain peer p_3's information from peer p_2, so peer p_1 knows about peer p_3 but the reverse might not be true). Notice that if a PPLive overlay is an undirected graph, all our results in this paper remain unchanged except the Figure 3.

k response degree. We call the size of a node's partner list as the node degree. One difficulty in obtaining the partner list (via the PPLive API) is that successive queries to the same node may yield slightly different partner lists. Since PPLive is close source, it is difficult to tell if the node returns only the subset of its partner list or the entire list of partners or some random partners, or if the partner list is really changing over time. Hence, we need to define a notion of node degree or partner list that is generic and covers all possibilities.

We define the *k response degree of a node* as the aggregated set of partners returned in the first k responses from a node that is sent successive queries for its partner list. In our experiments, obtaining the first 15 responses ($k = 15$) from a node typically took up to 15 seconds. To verify whether the aggregated set of partners returned in $k = 15$ responses is sufficient, we select a set of 50 random peers attending one PPLive channel called *Channel 3* and send partner queries to them. Figure 1(b) shows that $k = 15$ obtains more than 90% of partners obtained by $k = 20$. Moreover, when $k > 15$, the curve in Figure 1(b) flattens out. Therefore, we use a default setting of $k = 15$ for our partner discovery operation in Section 3. However, we verify the generality of our experimental results for smaller values of k as well ($k = 5$ and $k = 10$). Notice that the choice of $k = 15$ is also to reduce the number of stand-by partners in the returned packets, which leave the channels but may still be returned for the partner queries. For a larger k, it is possible that we might retrieve more stand-by partners. Henceforth, in this paper, the terms *node degree*, *k response degree*, and *k-degree* are used interchangeably.



(a) Responsive peers in two PPLive channels



(b) 10 machines running the crawler

Fig. 2. Responsive peer ratio and number of machine running the crawler

2.4 Active Peer

The next challenge is to clearly define when a peer is considered an active peer, which is a part of a given overlay. This is complicated because one PPLive peer may simultaneously attend multiple overlays, including non-subscribed overlays. Further, some clients may be behind NATs or firewalls, and may not respond to a direct probe message. Thus, given an overlay G and a peer v, v is considered to be an *active peer* in G if either v appears in the membership list for G at the membership servers, or v appears in the partner list of some other active peer u. Notice that the definition is recursive. Formally, we define the predicate:

$ACTIVE(v, G) = \{v \in \textit{Membership Server List for } G\}$ OR $\{\exists u : ACTIVE(u, G)$ $AND\ v \in u.PartnerList(G)\}$

Our above definition also includes "silent" peers that may be behind firewalls or unresponsive. Even though we have not described our crawler yet (see Section 3), we need to justify the definition. We quickly present a simple experiment below to do so. We measured the fraction of peers that were captured by our crawler (see *Snapshot Operation* in Section 3), using the above definition of active peers (# of total peers), and responded to the protocol ping (# of responsive peers). Figure 2(a) shows the fractions for two different PPLive channels, Channel 1 and 2, over the course of 24 hours. The authors of [Hei et al. 2007] reported that around 50% PPLive nodes may be behind NATs. Since Figure 2(a) shows that more than 50% of the captured peers are non-responsive: it is important to consider the characteristics of these PPLive peers as a part of the overlay, and our definition does this.

3. STUDY METHODOLOGY

We name our crawler PPCrawLive, which has been in use since February 2006. We shared PPCrawLive's crawled traces and released the code as an open-source software since April 2008 [PPC]. We describe below the design of PPCrawLive.

We use Ethereal [Eth] to trace traffic between a PPLive peer and PPLive servers, and traffic among PPLive peers. Having understood these traffic patterns, we implement PPCrawLive in the socket level using the *UDP* transport protocol. PPCrawLive runs on a Linux machine (either a machine in our cluster at UIUC, or a PlanetLab node) and joins a given PPLive channel whose ID is feed as an input argument to the crawler (each channel has a unique ID). Essentially, PPCrawLive works the same as the client in Figure 1(a) but it does not perform step 1 because the channel ID is input. PPCrawLive consists of two operations: Snapshot Operation and Partner Discovery.

Snapshot Operation. To obtain all active peers attending a given channel, this operation works as follows. First, given the channel ID, the initiator requests the initial peer list from the PPLive membership servers (step 2 in Figure 1(a)), and uses this to initialize a local list denoted as L. Second, the initiator continuously scans L in a round-robin fashion, by sending a request for the partner list to each entry (step 3 in Figure 1(a)), and appending to L new peers (i.e., ones that it has not heard about before) received in the partner list replies. Third, when the initiator has received fewer than n new peers among the last Δ peers received, the snapshot operation terminates. This is because different PPLive channels have different sizes, and the size of one channel varies very much over a day. If the snapshot operation stops after a fixed amount of time, it may not obtain the entire population of the crawled channel if this channel is big. So, the termination when few new peers are found, works well for the variation of channel size. In our experiments, for most channels, we use $n = 8, \Delta = 1000$, for a channel with less than 1000 peers, we use $\Delta = 500$. With this setting, the snapshot operation typically takes between 3 and 8 minutes. To avoid flooding the network with our ping messages, new snapshot operations are initiated only once every 10 minutes.

We define the *channel population size* as the number of active peers captured by one execution of the snapshot operation. We use the terms channel population size, channel population, and overlay size interchangeably.

Partner Discovery. This operation obtains the k response degree of a node as defined in Section 2.3. In our experiment, to obtain k responses from one peer p, we send $(k+2)$ successive requests to p for its partner list (e.g., we repeat step 3 in Figure 1(a) $(k+2)$ times for peer p). The first k received responses are aggregated to create the k response degree.

Essentially, there are two design choices - either to obtain each node's k response degree or to quickly crawl the entire overlay. We choose the former because we can almost instantly achieve the k-degree of nodes, which is critical to understanding the overlay characteristics of PPLive network. However, this may incur crawling lag when crawling the entire overlay. Particularly, to achieve the connectivity graph G (including nodes and links) of a given set of nodes, the partner discovery operation needs to travel from the first element to the last element of the set, for which it obtains the k-degree. This process incurs lag and thus G may not be an instant graph due to the high churn in PPLive overlays. In our experiment, we address the crawling lag by running parallel instances of PPCrawLive as presented below.

PPCrawLive is self-contained and easily parallelized. Each instance of PPCrawLive can be run independently in a machine. To increase the coverage of our crawler and reduce the impact of crawling lag, we run it simultaneously on multiple machines. Figure 2(b) shows the number of captured peers with m machines as a fraction of the number of captured peers with 20 machines (at four different times in a day). We observed that 10 machines cover about 98% of peers covered by 20 machines. Hence, we use 10 geographically distributed PlanetLab nodes to run simultaneous crawlers. We select PlanetLab nodes because of their worldwide distribution.

Name	Channel Size (Aggregated for a day)	Channel Type
A	32K-45K	Movie
B	10K-15K	Cartoon
C	8K-12K	Movie

Table I. Channels A, B, and C were studied from 02/2006 to 12/2006. In 2007 and 2008, we studied 37 other channels including sports, live TV, movies, and fashion channels.

	Studied Characteristics	Characteristic Type
1	Node degree distribution	Node Level Overlay Characteristics
2	Randomness of overlay	
3	Node's session length	
4	Peer availability interdependence	Inter-node Overlay Characteristics
5	Locality-awareness of overlay	
6	Channel population size	System-wide Overlay Characteristics
7	Participation in simul. overlays	
8	Resilience of overlay	

Table II. Studied Characteristics of the PPLive IPTV system

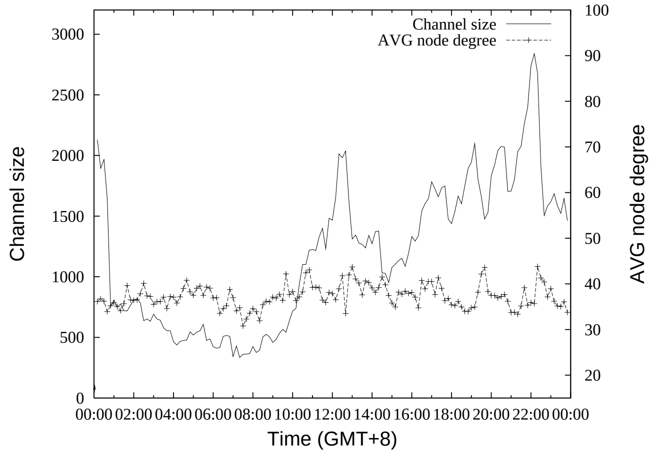
Notice that we synchronize the partner discovery and snapshot operation in our experiments. Particularly, the crawling interval is 10 minutes to avoid flooding PPLive overlays with our crawling messages. For each crawling interval, we start both snapshot operation and partner discovery. For the partner discovery, requests are sent to a node p successively until k responses are received from p. At that moment, no more requests are sent to p. In other words, when $k = 15$, it takes up to 15 seconds to obtain p's partner list and after that p will not be probed by the partner discovery for the rest of the crawling interval. We understand that the crawled data may be different if we vary the crawling interval as discussed in [Stutzbach et al. 2008]. However, finding the optimal crawling interval is not the focus of this paper. Instead, it could be a direction of future work.

Studied Channels. In our previous reports [Vu et al. 2006; 2007], we focused on three channels as shown in Table I. For anonymity, we name these channels as A, B, and C. Out of these, A is the most popular channel, C is the least popular channel, while B is somewhat in between A and C. Since 2007, we have studied 37 other channels including sports, entertainment, games, live TV, movies, stock market, and fashion channels. Since a large fraction of PPLive users is in China, we use the Chinese Time Zone (GMT+8) in our plots.

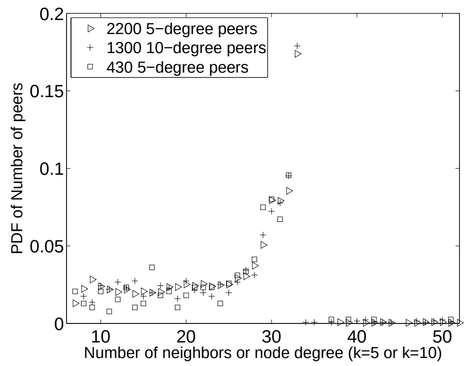
In the following sections, we present our findings and discussions about PPLive overlay characteristics from the view of node level, inter-node level, and system-wide level overlay characteristics as shown in Table II. Where possible, we compare and contrast our findings with the well-known overlay characteristics of p2p file-sharing [Ripeanu et al. 2002; Saroiu et al. 2003; Bhagwan et al. 2003].

4. NODE LEVEL OVERLAY CHARACTERISTICS

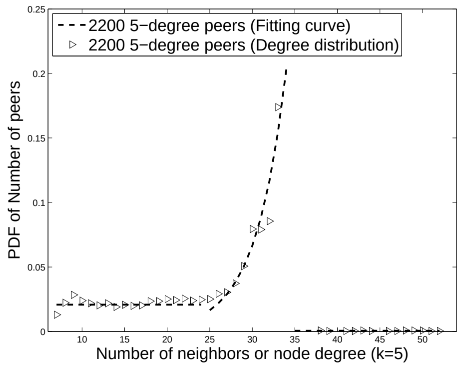
In this section, we study the overlay characteristics of the PPLive streaming network from the view point of a single node. Concretely, we model the node degree distribution, characterize the randomness of PPLive overlays, and model the session lengths of peers attending PPLive overlays.



(a) AVG node degree is independent of channel size (12/2006)



(b) Node degree dist. (05/2008)



(c) Node degree dist. fitted by Matlab

Fig. 3. Characterizing and modeling the node degree distribution

4.1 PPLive Overlay Structures are Similar to Random Graphs

It is well-known that the node degree distribution in p2p file-sharing networks is scale-free and hence likely a small-world network [Ripeanu et al. 2002; Saroiu et al. 2003]. This section shows that like p2p file-sharing overlays, the average node degree in the PPLive overlay is also independent of the channel population size. However, unlike p2p file-sharing overlays, the structure of PPLive overlay turns out to be closer to that of random graphs.

4.1.1 *Average Node Degree is Independent of Channel Population Size.* We simultaneously ran the snapshot operation to obtain active peers attending the channel A, and partner discovery to obtain the node degree of 300 randomly selected peers attending channel A, considering both active and responsive peers. Figure 3(a) shows the variation of the average node degree and channel population size of channel A during a day (i.e., 24 hour period). We first observe that although the average node degree varies, it stays within a small range - between 28 to 42 over the course of the day. More interestingly though, *there appears to be no correlation between the variation of average degree and the channel size.* This might be be-

Set of Peers	a	b	c	d	p	q	u	v	t
2200 5-degree peers	0.0228	$1.54\cdot 10^{-5}$	0.28	0.0006	7	24	33	38	52
1300 10-degree peers	0.0213	$8.14\cdot 10^{-6}$	0.3	0.0012	8	24	33	34	51
430 5-degree peers	0.0181	$4.26\cdot 10^{-6}$	0.33	0.0026	7	24	33	37	51

Table III. Parameters in Eq. 1 obtained from a piecewise function fitted by Matlab

cause regardless of the channel size, a PPLive needs a certain number of partners to maintain the streaming quality. In our experiments, we observe similar behavior for other studied channels.

Node Degree Distribution Model. To understand the node degree distribution, we ran the partner discovery on three channels and plot the distribution of the node degree in Figure 3(b). In this figure we observe that the node degree lies between 7 and 52. We also observe that in the two ranges from 7 to 25 and from 34 to 52, the node degree distribution exhibits a uniform distribution. In between, in the range from 25 to 33, the node degree indicates an exponential increase. Moreover, about 50% of peers has their node degrees between 28 and 33, while a very small number of peers have their node degrees greater than 34. Formally, we model the node degree distribution in Figure 3(b) using the following piecewise function:

$$y = f(x) = \begin{cases} 0 & if \;\; x < p \; or \; x > t \\ a & if \;\; p \le x \le q \\ b \cdot e^{c \cdot x} & if \;\; q < x \le u \\ d & if \; v \le x \le t \; and \; u < v \end{cases} \tag{1}$$

In Equation 1, x denotes the node degree ($x > 0$) and y denotes the probability that a peer has x neighbors ($0 \le y \le 1$). a,b,c, and d are positive. p, q, u, v, and t represent the limit parameters where the node degree distribution changes its behavior. Figure 3(c) shows the piecewise fit obtained from Matlab (or function y in Equation 1) for one channel. Correspondingly, Table III gives the coefficients fitted by Matlab and parameters for three channels. Here, the maximum sum of square errors of the fits is $2\cdot 10^{-3}$. It turns out the values of coefficients a, b, c, and d are fairly consistent for these channels. Therefore, we believe the piecewise fit approximates very well the real node degree distribution. We verify: $a \cdot (p - q) + \sum_{x=q+1}^{u} b \cdot e^{c \cdot x} + d(t - v) \simeq 1.0$ with coefficients and parameters in Table III.

It is clear that the node degree distribution consists of two main distributions uniform and exponential. The uniform distribution holds for the ranges of [7,24] and [34,52]. The exponential distribution is in the range of [25,33]. Since neither of these two distributions is heavy-tailed, we conclude that the node degree distribution is not heavy-tailed. In other words, PPLive overlays are not power-law graphs. This finding is similar to the finding made by Wu et al. [Wu et al. 2007b]. In particular, they find that the distributions of total number of partners at the stable peers in the UUSee network do not follow power-law distributions.

4.1.2 *Randomness of Overlays May Depend on Channel Population Size.* The distinction between a random and a non-random graph can be quantified by the metric called Clustering Coefficient (CC) [Watts and Strogatz 1998]. Informally, the CC metric of a graph is defined as follows: for a random node u and two

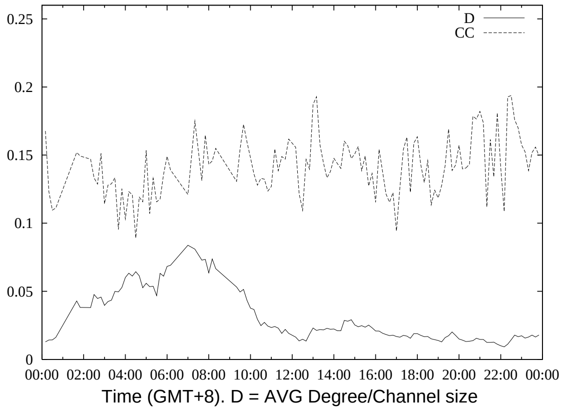


Fig. 4. Overlay resembles a random graph when channel size is small but becomes more clustered when channel size grows. For this figure, $k = 5$. (12/2006)

neighbors v and w selected randomly from u's partner list, CC is the probability that either v is in w's partner list, or vice versa. Notice that CC for a random graph is the average node degree divided by the system size (number of nodes).

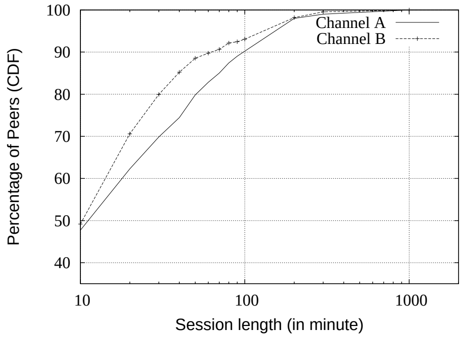
For our experiment, we first calculate the average degree of the PPLive overlay measured by the partner discovery operation, and calculate the metric D, the unconditional probability that v links to w:

$$D = (Average\ node\ degree)/(Channel\ size) \tag{2}$$

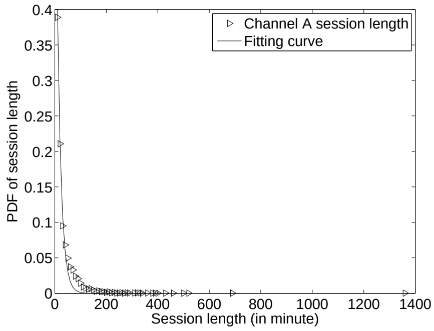
We then compare D to CC, which is measured as follows. In each snapshot, we randomly select a set S of 300 responsive peers of the channel A. For a peer p in S, we first use partner discovery to obtain its partner list. Second, we randomly pick two responsive partners p_1 and p_2 in p's partner list and obtain their partner lists using partner discovery. Third, we verify whether p_1 is in p_2's partner list or not, or vice versa. If p_1 is in p_2's partner list (or vice versa), we increase the variable called $Count$ by 1. $Count$, initialized to 1, represents the total number of edges existing in all such partner pairs. Then, CC is computed as follows:

$$CC = Count/(2 \times ResponsiveNodeNum) \tag{3}$$

In Equation 3, $ResponsiveNodeNum$ is the number of active nodes whose two active partners p_1 and p_2 are verified (i.e., $ResponsiveNodeNum = 300$ in this experiment). Figure 4 plots the 24-hour variation of D and CC for $k = 5$. This experiment was done at the same time and for the channel A as shown in Figure 3(a). We observe that generally when the channel population size is small, the value of CC is close to the value of D (e.g. 4AM-8AM period). This indicates that *when channel population size is small, the structure of the PPLive overlay graph approaches a random graph.* As the channel population size increases (10:00 AM onwards in Figure 4), the CC grows to about six times that of the value of D. This is still indicative of some randomness of the graph, although it is clear that larger channel population sizes lead to more clustering. We verify this behavior with $k = 10$ and $k = 15$ in our previous work [Vu et al. 2006]. In [Wu et al. 2007b], authors find that the graph of *stable peers* attending the UUSee network exhibits



(a) CDF of session lengths. The X-axis is on a log-scale. (12/2006)



(b) PDF of session length fits a geometric series. (12/2006)

Fig. 5. Characterizing and Modeling the session length distribution

small-world properties (more clustered than random graphs) and the overall UUSee network may represent a low network diameter.

4.2 PPLive Peers are Impatient

It has been widely reported, e.g., [Saroiu et al. 2003], that users of p2p file-sharing systems are "patient," i.e., they do not mind waiting hours, and sometimes even days, for file downloads to complete. In the PPLive environment, due to the streaming nature of the content, the opposite is true. In other words, PPLive users are very impatient in terms of staying in a particular channel. They usually switch channels during their watching time.

Figure 5(a) shows session lengths of 5000 random peers taken from 38675 peers in channel A, and 5000 random peers taken from 11625 peers in channel B. We observe that about 50% sessions are shorter than 10 minutes, 60% of A's sessions and 70% of B's sessions are shorter than 20 minutes, and over 90% sessions from both channels are 100 minutes or shorter. This implies that *PPLive nodes are impatient*, i.e., they rarely stick to a channel for too long.

This behavior arises out of both a difference in application characteristics, as well as from user behavior. Since p2p file-sharing overlays like Kazaa are *batch-mode* delivery systems in which the human users can go away from the client machine while it continues to download content, session lengths tend to be long. In comparison, the PPLive application is a *streaming-mode* system, where a user can obtain benefits from the application only if she is actively present near the client machine. If the user is not at her machine, she has a lower incentive to keep her PPLive client running compared to p2p file-sharing system, hence the session times are shorter.

There are other reasons contributing to the short session lengths. First, PPLive users are likely to switch from one channel to another because of a loss of interest - home television viewing often suffers from the same malady! Second, PPLive nodes face a longer start-up delay than nodes in p2p file-sharing systems. We have observed that newly joining nodes need tens of seconds to a minute to join a channel, with the latency being even higher if the channel is really small (due to the scarcity of potential neighbors). This long start-up delay increases the likelihood of the user switching to a different channel.

Session Length Model. To understand properties of PPLive peers' sessions, we use Matlab to model the PDF of session lengths. Since PPCrawLive runs every 10 minutes (Section 3), node's session lengths were measured only multiples of 10-minute periods. Thus an appropriate model would be a *discrete* mathematical series, rather than a continuous distribution. Figure 5(b) shows fitting curve obtained from Matlab for the channel A. While the fitting curve is an exponential function of time (since Matlab offers only continuous fits of data), we express the session length distribution as the (equivalent) geometric series.

Concretely, the geometric series can be expressed as follows. Let y be the probability that a node's session length is measured as $x \cdot 10$ minutes (where $x > 0$). Our models reveal the relationship between y and x as:

$$y = a \cdot e^{10 \cdot b \cdot x} \tag{4}$$

Here, a and b are constants. a is the base of the geometric series, and the multiplicand in the geometric series is $r = e^{10 \cdot b}$. Factor 10 in the above equation arises from our discretized session lengths that are multiples of 10 minutes.

Channel	**a**	**b**
A	0.6378	-0.05944
B	1.183	-0.09878
C	1.079	-0.09594

Table IV. Coefficients of geometric series with $y = a \cdot e^{10 \cdot b \cdot x}$, fitted by Matlab.

Table IV shows values of a and b obtained by fitting the session lengths of three channels A, B, and C to continuous exponential curves in Matlab. Here, the corresponding sums of square errors of the fits vary from $1.5 \cdot 10^{-4}$ to $2 \cdot 10^{-4}$. We verified that this indeed leads to the geometric series by verifying, for each channel, that the value of $\Sigma_{i=1}^{\infty} a \cdot r^i$ turned out to sum to 1.0.

In conclusion, the application characteristics and user behaviors cause *very short session lengths and consequently a higher degree of churn in PPLive than in p2p file-sharing overlays.* Our model of geometrically distributed session lengths of nodes (per channel) can be used to accurately model node arrival/departure behavior in simulations of media streaming p2p systems. This can be used to improve the believability of simulation set-ups for media streaming p2p systems by using realistic modeled workloads. This also opens up an opportunity of incorporating session-length-based optimizations at run-time in real deployments. Finally, our model of geometrically distributed session length times indicates a high degree of *homogeneity* across nodes in the session lengths, and this indicates that homogenous protocol designs have substantial promise and are a good match for media streaming p2p overlays - this does not of course preclude benefits from heterogeneous protocol designs. Future designs for both streaming p2p overlays and generic p2p routing substrates will have to keep these issues in mind.

5. INTER-NODE OVERLAY CHARACTERISTICS

In this section, we study the overlay characteristics of the PPLive network from the view point of a pair of nodes. In particular, we characterize peer availability interdependence and the locality-awareness of PPLive overlays.

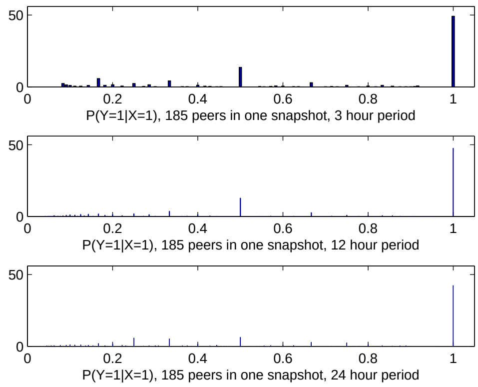


Fig. 6. Peers occurring in the same snapshot may occur together again. Plot shows PDF of availability correlation. Y-axis is % host pairs. (12/2006)

5.1 Peer Availability Interdependence

P2P file-sharing systems are known to have host uncorrelated availabilities [Bhagwan et al. 2003]. In comparison, we show that: (1) unlike in p2p file-sharing systems, PPLive peer pairs occurring together in a snapshot have highly *correlated* availabilities, while (2) like in p2p file-sharing systems, peer pairs that are randomly selected from different snapshots will have highly *uncorrelated* availabilities.

We measure the correlation between the availability of two peers X and Y by using a similar technique as in [Bhagwan et al. 2003]. Specifically, let $X = 1$ (resp. $Y = 1$) be the event that the peer X (resp. Y) occurs as an active peer in a given snapshot. Then, for the peer pair (X, Y), we calculate $P(Y = 1|X = 1)$, i.e., the conditional probability that given X is present in a given snapshot, Y will be too. We then compare this conditional probability to the unconditional probability that peer Y occurs in a given snapshot, i.e., $P(Y = 1)$. The closer the two values, the more uncorrelated are X's and Y's availability patterns.

5.1.1 *Nodes in the Same Snapshot Have Correlated Availability.* Given traces of a series of snapshots (for Channel A) taken over a contiguous time period (we use three settings: 3 hours, 12 hours, and 24 hours), we select a set of 185 peers from the first snapshot at 12AM (starting of a day). Notice that we have 144 snapshots for 24 hours. Figure 6 shows the conditional probability $P(Y = 1|X = 1)$, for each node pair in this set. 50% of node pairs show a high correlation in availability, i.e., $P(Y = 1|X = 1) = 1$.

We believe there are two factors contributing to this behavior: first, user pairs that appear in the same snapshots are likely to have similar interests in terms of channel viewing contents and viewing time. Second, and perhaps more importantly, certain peer pairs that occur together in a snapshot are perhaps "well-matched" as streaming relays for each other. It is likely that PPLive's inter-overlay optimizations (similar to Anysee system's inter-overlay optimizations [Liao et al. 2006]) cause one client's presence to draw in other well-matched clients for relaying. In our experiment, we observe the same results with channel B and C.

5.1.2 *Random Node Pairs Have Independent Availabilities.* We ran a similar experiment as in Section 5.1.1, except that we selected 500 random peers from among 39412 peers crawled over 24 hours from channel A, as well as 500 random peers from 11527 peers crawled over 24 hours from channel B. Then, we computed the difference between $P(Y = 1|X = 1)$ and $P(Y = 1)$ for each host pair (among the set of 500) over the 144 snapshots, corresponding to 24 hours. In contrast to results in Section 5.1.1, Figure 7 shows that random peer pairs have completely independent availability behavior. In particular, 87% peer pairs in channel B (92% in channel A) lie between +0.2 and -0.2, indicating independence in availability among these peers. This is explainable because random peers are unlikely to have either correlation in user interests (i.e., viewing time, viewing content) as peers in the same snapshot, or be well-matched in relaying feeds.

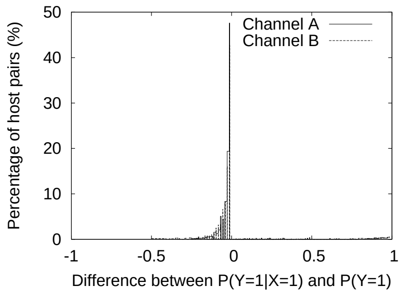


Fig. 7. Randomly selected pairs of peers have uncorrelated availabilities. (12/2006)

In conclusion, unlike p2p file-sharing systems, media streaming p2p systems may exhibit a higher correlation availability among certain node pairs. Systems designers will have to account for this, regardless of whether it arises from user interests or from internal optimized design of the PPLive overlays (in the latter case it is a good p2p system design principle).

5.2 PPLive Overlay is Slightly Locality-aware

This section evaluates the effect of locality in choosing PPLive streaming partners. We first study the distance between pairs of neighbors in PPLive overlays. Second, we render the topology of PPLive overlays with nodes and links. This rendered graph gives more insights about the overall connectivity of PPLive overlays.

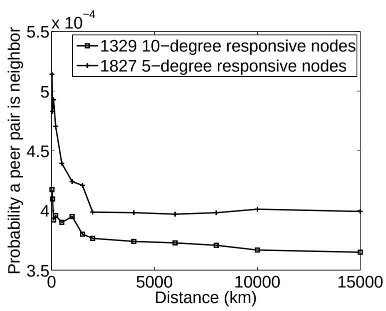
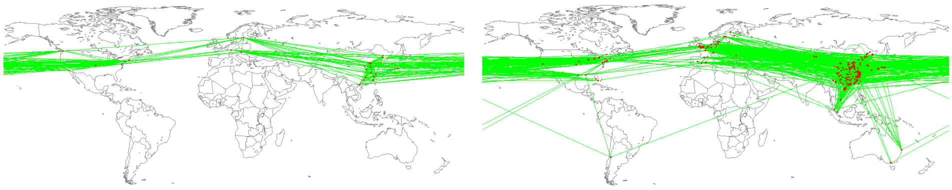


Fig. 8. Closer peers have a slightly higher probability to be neighbors. (05/2008)



(a) One overlay of 70 nodes (b) One overlay of 4000 nodes

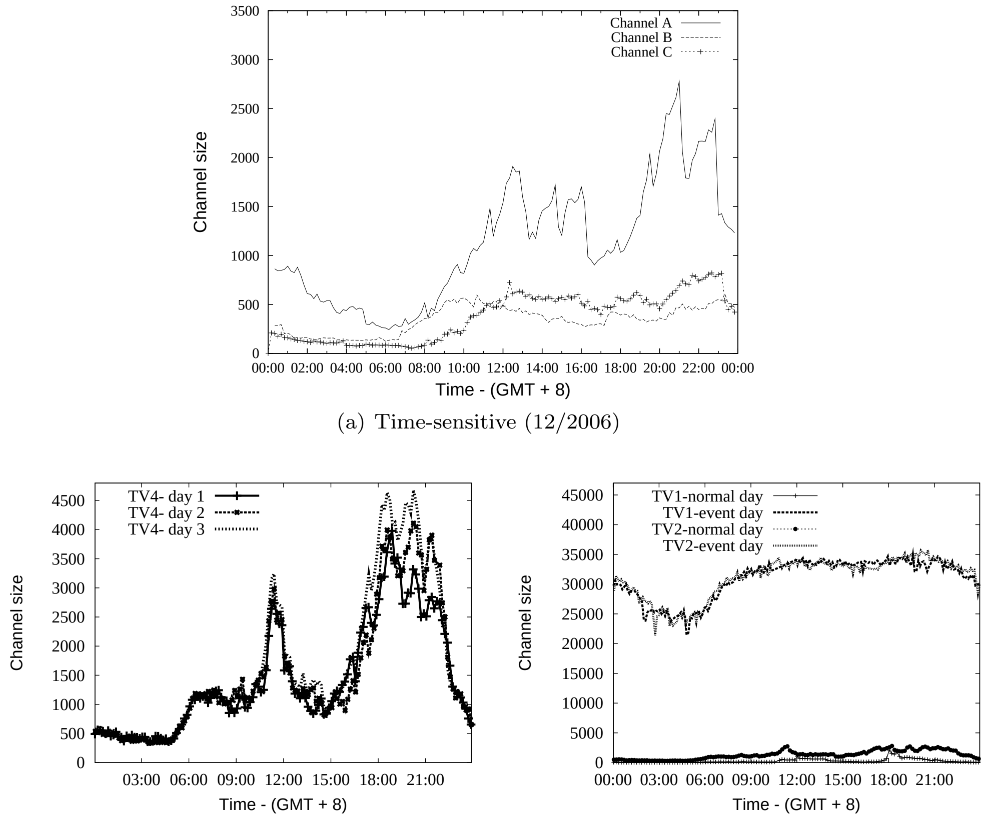
Fig. 9. Rendering the PPLive topology. Nodes fall into three main regions: China, Europe, and North America (05/2008)

5.2.1 *Geographically Close Peers are Likely to be Neighbors.* In this experiment, we collect two sub-overlays with 1329 and 1827 random peers, respectively. The former consists of nodes with 10-degree and the later consists of nodes with 5-degree. We perform following steps to obtain the distance between peer pairs in the two above sub-overlays. First, we obtain the Longitudes/Latitudes, based on peers' IPs, from MaxMind database [Max]. Second, given the Longitudes/Latitudes, the distance in kilometer is computed according to the Haversine formula [Hav]. We use the geographical distance to approximate the network locality because although they are not equivalent, the geographical distance is a good approximation of the network locality (delay), which can not be measured accurately by all means.

Figure 8 shows the relationship between probability that a random peer pair are partners and distance between the peers. This figure indicates that if the distance between two peers is less than 2000 (km), they have a slightly higher probability to be neighbors, independent of distance. In contrast, peer pairs that are between 2000 (km) and 15000 (km) have a nearly the same probability to be neighbors.

There are two possibilities for this behavior. First, there is no locality-awareness in choosing PPLive streaming partners. This is because a very large portion of PPLive peers is in China (i.e., more than 80%). So, although peers in China choose streaming partners at random, it is likely that a peer in China will choose peers in China as its partners. Thus, the distance between peers may be close although this partner selection is random. Second, PPLive peers take the geographical location into account in choosing streaming partners. In other words, when selecting neighbors, a PPLive peer chooses geographically closer peers. In this case, a peer in China may still choose a partner in China but this selection is locality-aware. However, notice that in Figure 8 geographical locality provides only about 10% high probability of partnering. This arises from the randomness of PPLive overlays.

5.2.2 *Rendering the PPLive Overlays.* To understand the 2000 km cut-off in Figure 8, we visualize two overlays obtained from two snapshots of 70 and 4000 nodes by Geoplot [Geo] in Figure 9. In this figure, peers fall into three main clusters in China, Europe, and North America, where peers within one cluster connect to each other. However, there exists a large number of links across these clusters, especially links to/from the China cluster. The fraction of links to/from the China cluster over the total number of links in the overlay is higher in a smaller overlay. More interestingly, the diameter of each cluster (China, European, or North America) is roughly about 2000 (km). This might explain the cut-off in Figure 8. In other words, nodes within a cluster are slightly more likely to create partnerships



(a) Time-sensitive (12/2006)

(b) Self-repeated (05/2008)

(c) Event-dependent (05/2008)

Fig. 10. Channel size is time-sensitive, self-repeated, and event-dependent.

for video streaming, but many links exist across the main clusters.

Section 4.1 suggests that PPLive overlays might be random graphs. This section shows that the partnership of PPLive peers is slightly locality-aware. These two results confirm that overall the PPLive overlays are similar to random graph in structure but there exists a small degree of locality awareness.

6. SYSTEM-WIDE OVERLAY CHARACTERISTICS

This section studies the overlay characteristics of the PPLive network from the system-wide level. Specifically, we focus on the channel population size, peer participation in simultaneous overlays, and the resilience of PPLive overlays to the massive failure of nodes.

6.1 Channel Population Size is Time-sensitive, Self-repeated, and Event-dependent

Studies on p2p file-sharing systems [Bhagwan et al. 2003] showed that diurnal patterns and churn exist, but the size of a p2p overlay stays stable in spite of these features. The findings in this section show that (1) PPLive overlays have a highly variable channel population size (as well as high churn and diurnal patterns), (2) the channel size exhibits self-repeated behavior over days, and (3) the channel size changes suddenly when the real-world events occur.

We first study the time variation of channel population size of PPLive channels. Figure 10(a) shows the variation of the channel size for the three PPLive channels A, B, C over the course of a day. We observe that all channels have peak populations at noon and evening/night, and are smallest in the early morning. This might be because users usually use PPLive in spare time (at noon and evening/night).

The second study reveals that the PPLive channel size is self-repeated as shown in Figure 10(b). Particularly, we study a live TV channel for three random and normal days (the days without any special public events). The channel variation follows the same pattern for all the four days with peaks at noon and night, and becomes smallest in the early morning. This confirms that the channel size variation of PPLive channels is self-repeated and consistent for normal days.

In contrast, the channel size shows a sudden increase during a special event. While we were conducting our experiments, the Great Sichuan earthquake occurred in China in May 2008. We happened to measure two live CCTV channels during this period. Figure 10(c) shows the the channel size variation during the course of a day, both before and right after the earthquake. Before the earthquake, the channel size was less than 5000 and time-sensitive. However, right after the earthquake the channel size increased dramatically to about 35000 users, resulting in a flash crowd. More interestingly, although the channel sizes was smallest in the early morning, the peaks at noon and night disappeared, and the channel size remained high after 9AM. This flash crowd might be because during the earthquake period, there were many people both inside and outside China watching PPLive channels for the news of the earthquake and thus the channel size stayed high. We observed that the channel sizes remained high for two weeks after the earthquake. That means, events can trigger a large population of viewers to the usage of p2p streaming systems. This is consistent with the increase of viewers during Chinese New Year event [Hei et al. 2007], World Cup Soccer Games [Silverston and Fourmaux 2007], or the mid autumn festival in China [Wu et al. 2007b].

In conclusion, the PPLive channel size distribution is time-sensitive, self-repeated and event-dependent. Understanding this behavior is important for network planing. For example, designers can place more proxies to relay streams when the channel size is small, or when an event occurs, thus reducing the startup latency and minimizing the churn.

6.2 Peer Participation in Simultaneous Overlays Follows the Zipf Distribution

The PPLive system is widely believed to use the idea of inter-overlay optimizations, which is used in the Anysee system [Liao et al. 2006] and Skype system [Suh et al. 2006]. As a result, a client machine may appear as a participant in multiple overlays, including ones that the user is not subscribed to. In this section, we study peers attending multiple channels (overlays) simultaneously, which we call *interoverlaying peers*. Particularly, we crawl 35 simultaneous channels, chosen at random, and extract interoverlaying peers. At the same time, we probe these interoverlaying peers to obtain those which are responsive to PPLive protocol pings and call them *responsive interoverlaying peers*. Figure 11 shows the distributions of interoverlaying peers and responsive interoverlaying peers at four different time stamps in a day. Notice that this figure is in log-log scale. For example, at 3PM

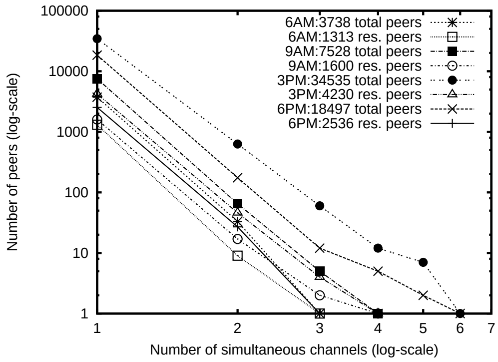


Fig. 11. Peer Participation in Simultaneous Overlays Follows the Zipf Distribution. Notice that the plot is in log-log scale. Time GMT+8. (05/2008)

we collected 34535 peers from 35 channels, among these peers 4230 peers are responsive. We then count the number of interoverlaying peers from 34535 peers, the number of responsive interoverlaying peers from 4230 peers, and plot these two counters in Figure 11. This figure indicates that the distributions of both interoverlaying peers and responsive interoverlaying peers follow a Zipf distribution. There might be different reasons for this Zipf distribution. For example, the Zipf distribution might result from the participation of peers in simultaneous overlays. The Zipf distribution might also result from the *crawling lag* of our crawler, which runs every 10 minutes. However, based on our extensive experiments, we *observationally* conclude that PPLive peers join simultaneous overlays as discussed below.

First, as we presented in Section 4.2, the session length of PPLive peers follows a geometric distribution. Moreover, the crawling interval in our experiments is 10 minutes. Thus, for a peer p, the probability that p attends c channels during this 10-minute period would be the Poison distribution instead of a heavy-tailed distribution as shown in Figure 11. This means that session times do not solely contribute to Figure 11 and the *crawling lag* of our crawler would not give the Zipf distribution in Figure 11. Therefore, our hypothesis is that Figure 11 results because a PPLive peer may join simultaneous PPLive overlays.

Second, the existence of a large number of responsive interoverlaying peers indicates that the interoverlaying peers might not be proxies, which relay the streaming traffic. Instead, they might be real PPLive client machines. That means PPLive might have an internal mechanism to leverage peers so that they can share their available resources to support peers in non-subscribed overlays, which differ from their subscribed overlays.

Finally, although the maximum number of simultaneous channels a peer can attend is 6, we have crawled a very large data set of 35 simultaneous channels over a long period of time and we have observed the consistent Zipf distribution in Figure 11. We thus *observationally* conclude (rather than deterministically) that PPLive peers might join simultaneous channels. Together with the fact that PPLive is a close source system and the insight protocol is unknown, our conclusion reflects our best knowledge and all information we can obtain from the PPLive system.

To further understand the Zipf distribution, we fit the curves in Figure 11 with the function $y = a \cdot x + b$ in Matlab. Table V lists the coefficient a, or the θ

parameter of the Zipf distribution. We observe that the values of θ are comparable for all curves, consistent with the similar slopes of the linear fit $y = a \cdot x + b$. This means the distribution of interoverlaying peers remains consistent over time.

Data Set	θ
6AM:3738 total peers	-6.355
6AM:1313 responsive peers	-6.606
9AM:7528 total peers	-6.48
9AM:1600 responsive peers	-6.135
3PM:35535 total peers	-5.049
3PM:4230 responsive peers	-6.358
6PM:18497 total peers	-5.745
6PM:2536 responsive peers	-6.052

Table V. Coefficients of the linear fit with $y = a \cdot x + b$, fitted by Matlab. ($\theta = a$).

Given the large data set obtained in this experiment and the consistent values of the θ as presented in Table V, we conclude that PPLive peers might join multiple overlays at the same time and the distribution of peer participation in simultaneous overlays follows the Zipf distribution.

6.3 Resilience of PPLive Overlays

It is well known that the overlay connectivity of p2p file-sharing networks is power-law distributed and the node degree distribution follows the Zipf distribution [Ripeanu et al. 2002]. In p2p file-sharing overlays, a few nodes in the network have significant higher degree than the others. When these high degree nodes are under orchestrated attacks and broken, the overlay easily becomes disconnected. In this section, we are interested in the resilience of PPLive overlays in the face of failures or attacks. To do so, we set up the following experiment:

—Randomly select a set S of nodes currently attending a PPLive channel.

—Use partner discovery operation to obtain partner lists (i.e., k response degree) of all nodes in S. The partner list of a peer p in the set S consists of links from p to other nodes in the overlay.

—Remove all unresponsive nodes in S (i.e. those nodes that return no partners to our queries.) to obtain a set S_1. Notice that S_1 is a subset of S and each node in S_1 has a partner list.

—For each node p in S_1, scan all elements of p's partner list and obtain the subgraph G whose vertex set is S_1.

—Find the biggest connected component G_1 within G. This step is required because G might not be a connected graph.

After the above steps, we obtain a connected component G_1 of responsive nodes. By studying the connectivity of responsive nodes, we can infer the connectivity of the entire PPLive overlays. In our experiment, it turned out that the selected channel has 3218 nodes (the size of S is 3218) and G_1 has 1625 nodes. Figure 12 shows the node degree distribution of all nodes in G_1, in which the average node degree is 5.77. Notice that this average degree is significantly lower than the node degree in Section 4 because G_1 contains only responsive nodes and links between

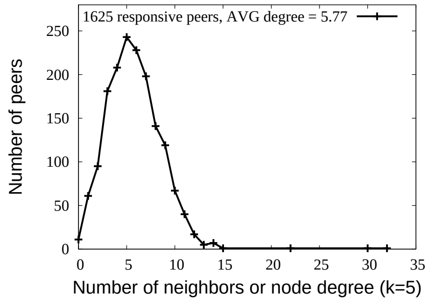


Fig. 12. Node degree distribution of a connected component G_1. (05/2008)

them. The degree distribution of nodes in G_1 is the Gaussian distribution with the standard derivation 2.81.

Next, we measure G_1's resilience. For this, we perform two different deletion strategies - these are called *highest degree deletion* and *random deletion*. For the first strategy, we recursively delete the node with the highest degree and all links from this node to other nodes. This is done until G_1 is disconnected. This deletion strategy is deterministic and when we delete 13 nodes, G_1 becomes disconnected. For the second strategy, we recursively delete a random node and all links from this node to other nodes in G_1. This is done until G_1 is disconnected. To remove the bias of the random node selection, we perform the second deletion strategy 100 times. Table VI compares the two deletion strategies. We observe that the mean and median of the number of deleted nodes obtained from 100 random deletions is not very different from the number of deleted nodes in the highest degree deletion strategy. Together with the node degree distribution in Figure 12, this table implies that the connectivity of G_1 is close to random and G_1 is loosely connected (G_1 becomes disconnected when fewer than 1% of nodes are removed from it).

Metrics	Random	Highest degree
Mean	16.3	13
Median	14	-
Standard Derivation	14.78	-
Min	1	-
Max	68	-
95 Percentile	41	-
5 Percentile	1	-

Table VI. Comparison between Random Deletion and Highest degree Deletion of G_1

It is well-known from previous studies that p2p file-sharing overlays are robust in the face of random massive failures but become vulnerable to orchestrated attacks due to their power-law natures [Saroiu et al. 2003]. In contrast, PPLive overlays are fairly random, since the random deletion results in the similar outcome as highest degree deletion (similar to orchestrated attacks). In other words, for an overlay with the same number of nodes and a similar node degree distribution, a PPLive channel overlay is more resilient to the massive failure of nodes than that of p2p file-sharing. This characteristic is likely related to the fact that maintaining a good streaming quality requires a more robust overlay structure, especially under a very high churn environment like the PPLive network.

7. RELATED WORK

It is well-known that the p2p file-sharing overlay is small world in nature [Ripeanu et al. 2002; Saroiu et al. 2003]. However, our study shows that the structure of PPLive overlay is closer to that of random graphs. Similarly, while p2p file-sharing systems are believed to have host availabilities uncorrelated, availability correlation of PPLive peer pairs varies in certain situations. Studies on p2p file-sharing systems also indicate that although churn exists, the size of a p2p overlay remains stable [Bhagwan et al. 2003]. In contrast, the PPLive overlay size varies significantly and peaks both at noon and during night. The channel population size is also event-dependent and increases dramatically during the event period. Moreover, users of p2p file-sharing are reported to be patient [Saroiu et al. 2003], while our study shows that PPLive users are relatively impatient.

Recently, a considerable number of measurement studies have been conducted for p2p IPTV systems such as PPLive [Vu et al. 2006; 2007; Hei et al. 2007; Silverston and Fourmaux 2007; Ali et al. 2006; Huang et al. 2008], PPStream [Silverston and Fourmaux 2007], Sopcast[Silverston and Fourmaux 2007], TVAnts [Silverston and Fourmaux 2006], CoolStreaming [Li et al. 2007; Li et al. 2008; Xie et al. 2007], and UUSee [Wu et al. 2007a; 2007b; 2008]. Except our preliminary reports [Vu et al. 2006; 2007] and the papers of Wu et al. [Wu et al. 2007b; 2008], which focus on the overlay characteristics of p2p IPTV systems, other projects mainly measure the network-centric or user-centric characteristics of the p2p IPTV systems. Particularly, the network-centric metrics have been studied such as peer churn rate, video traffic properties [Hei et al. 2007], throughput, video download policies [Silverston and Fourmaux 2007], rate of flow, duration of flow [Ali et al. 2006]. The user-centric metrics have also been investigated such as session length, user geographic distribution [Hei et al. 2007; Li et al. 2007], video buffering [Hei et al. 2007], throughput distribution [Wu et al. 2007a], user behavior, user satisfaction [Huang et al. 2008].

Our previous reports [Vu et al. 2006; 2007] and this paper focus on the overlay characteristics of the PPLive system. We have conducted a crawler-based study to measure and model the overlay characteristics of the PPLive network. Our crawler, PPCrawLive, which was implemented in February 2006 [Vu et al. 2006] and in parallel with the crawler used in [Hei et al. 2007], differs from the crawler used in [Hei et al. 2007] in two ways. First, their crawler runs for about 15 seconds every minute. Thus, to crawl a large part of the network, it imposes a high load on the PPLive system. In contrast, PPCrawLive runs every 10 minutes. Second, their crawler stops after a fix amount of time, regardless of the channel size while the stop condition of PPCrawLive depends on the crawled overlay size. Since April 2008, PPCrawLive [PPC] has been released as open source software to the research community worldwide.

After we conducted our measurement studies about the overlay characteristics of the PPLive system and published our report [Vu et al. 2006], Wu et al. studied the topology of a system named UUSee [Wu et al. 2007b; 2008]. Similar to what we had done with the PPLive overlays, they studied the node degree distribution and the randomness of UUSee overlays. Therefore, we are the first to study and publish results about the overlay characteristics of a p2p IPTV system. This paper significantly extents our preliminary reports on PPLive overlays [Vu et al. 2006;

2007]. To the best of our knowledge, we are also the first to provide mathematical models for the overlay characteristics of p2p IPTV systems.

8. DISCUSSION AND CONCLUSION

Results obtained from our extensive experiments indicate that PPLive overlay characteristics differ from those of p2p file-sharing. From our findings, we conclude that: (1) PPLive overlays are similar to random graphs in structure and thus more robust and resilient to the massive failure of nodes, (2) Average degree of a peer in the overlay is independent of the channel population size and the node degree distribution can be fitted by a piecewise function, (3) The availability correlation between PPLive peer pairs is bimodal, i.e., some pairs have highly correlated availability, while others have no correlation, (4) Unlike p2p file-sharing peers, PPLive peers are impatient and session lengths (discretized, per channel) are typically geometrically distributed, (5) Channel population size is time-sensitive, self-repeated, event-dependent, and varies more than in p2p file-sharing networks, (6) Peering relationships are slightly locality-aware, (7) Peer participation in simultaneous overlays follows a Zipf distribution. From these conclusions, we draw several lessons:

Lesson 1. PPLive peers slightly prefer to have topologically nearby partners and peers can attend simultaneous overlays, including their non-subscribed overlays. This improves the streaming quality of the entire system. Moreover, peers in the PPLive network fall in three main clusters in China, Europe, and North America with a large number of connections from/to the China cluster. Therefore, it is reasonable to strategically place stream relaying servers to support overlays, given that the overlay sizes are time-sensitive, self-repeated and event-dependent.

Lesson 2. Geometrically distributed session lengths of nodes can be used to accurately model node arrival/departure in simulations of media streaming p2p systems. Further, since the geometric distribution is indicative of *memoryless* session lengths (per node), this means that nodes are homogeneous w.r.t. their availability. Thus, *homogeneous* protocol designs for p2p overlays in this application space are reasonable. In other words, protocols that treat participating nodes equally are simpler and work effectively. This does not of course preclude benefits of heterogeneous protocol designs based on metrics such as bandwidth, CPU speed, etc.

Lesson 3. Our conclusion (1) indicates that small PPLive overlays work well by creating random overlay structures - thus, simple and homogeneous solutions work well at medium-scale (and not too large) channel sizes. Further, even when overlays are large, our conclusion (2) above indicates that homogeneous designs work well too. Notice that this does not preclude the use of heterogeneous protocol design.

Lesson 4. Since the availability correlations among node pairs are bimodal, this can be used to *fingerprint*, at run-time, which pairs of nodes are correlated and which are not. The bimodality of the behavior means that a few (random) sample points will suffice in categorizing each node pair as either “correlated” or “not correlated”. This availability information can then be used to create overlays that are either present all at once, or to route media streams (for a given channel) to a recipient node via other correlated nodes that are likely to also be up at the same time. This finding means simulations of media streaming p2p systems need to account for this bimodal availability correlation in the injected churn models.

Lesson 5. The structure of PPLive overlay is close to random. This randomness is to maintain the connectivity of the overlay and preserve the streaming quality under the high churn environment. Moreover, the random structure obtains the robustness and resilience to the massive failure of nodes. However, locality also needs to be taken into account in designing p2p streaming overlay so that the close peers have more chance to exchange stream and thus improve the streaming quality. Of course, extreme locality may create clustered overlays, which are vulnerable to the massive failure of nodes and churn. Therefore, designing a locality-aware p2p streaming system, which is resilient to churn and node failures, requires more attention and effort from research community.

Lesson 6. While measuring overlay characteristics of the PPLive network, we have faced numerous challenges and spent a significant amount of time to access the overlay due to its closeness. For future p2p multimedia streaming systems and online networks in general, there should be more accessible APIs so that the systems can be measured more easily and deeply. This helps researchers characterize the systems and thus can provide better suggestions to improve their performance. Otherwise, we need more efforts in designing open-source network crawlers like our PPCrawLive. **Notice that we share the PPCrawLive at "http://dprg.cs.uiuc.edu/downloads"**.

In conclusion, the differences between PPLive overlays and p2p file-sharing overlays drawn from our studies show that p2p systems designers may need to account for application nature. This study is also indicative of the challenge in designing "generic" p2p substrates catering to a wide variety of applications. Since custom-built substrates are wasteful, it may be important for system designers to address classes of p2p applications with common characteristics. Finally, a deeper study of user behavior (e.g., via HCI research) may yield novel p2p overlay design principles.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their helpful comments on improving the paper.

REFERENCES

Ethereal/WireShark homepage, http://www.wireshark.org/.

Geoplot, http://www.caida.org/tools/visualization/geoplot/.

Haversine formula, http://www.movable-type.co.uk/scripts/latlong.html.

Maxmind, http://www.maxmind.com/.

Multimedia research group inc, http://www.mrgco.com/.

Ppcrawlerlive: the crawler of the pplive system, http://dprg.cs.uiuc.edu/downloads.

PPLive homepage, http://www.pplive.com/.

Ali, A., Mathur, A., and Zhang, H. 2006. Measurement of commercial peer-to-peer live video streaming. In *Proceedings of Workshop in Recent Advances in Peer-to-Peer Streaming.*

Andersen, D. G., Balakrishnan, H., Kaashoek, M. F., and Morris, R. 2001. Resilient overlay networks. In *Proceedings of ACM Symposium on Operating Systems Principles (SOSP)*. 131–145.

Banerjee, S., Bhattacharjee, B., and Kommareddy, C. 2002. Scalable application layer multicast. In *Proceedings of ACM Sigcomm.*

Bhagwan, R., Savage, S., and Voelker, G. 2003. Understanding availability. In *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS).*

HEI, X., LIANG, C., LIANG, J., LIU, Y., AND ROSS, K. 2007. A measurement study of a large-scale p2p iptv system. *IEEE Transactions on Multimedia 9,* 8 (December), 1672–1687.

HUANG, Y., FU, T. Z. J., CHIU, D.-M., LUI, J. C. S., AND HUANG, C. 2008. Challenges, design and analysis of a large-scale p2p-vod system. In *Proceedings of ACM Sigcomm.* 375–388.

LI, B., QU, Y., KEUNG, Y., XIE, S., LIN, C., LIU, J., AND ZHANG, X. 2008. Inside the new coolstreaming: Principles, measurements and performance implications. In *Proceedings of IEEE Infocom.* 1705–1713.

LI, B., YIK, K., XIE, S., LIU, J., STOICA, I., ZHANG, H., AND ZHANG, X. 2007. An empirical study of the coolstreaming+ system. *IEEE Journal on Selected Areas in Communications, Special Issue on Advances in Peer-to-Peer Streaming Systems 25,* 9 (December), 1627–1639.

LIANG, J. AND NAHRSTEDT, K. 2006. Dagstream: Locality aware and failure resilient peer-to-peer streaming. In *Proceedings of ACM Multimedia Computing and Networking (MMCN).*

LIAO, X., JIN, H., LIU, Y., NI, L. M., AND DENG, D. 2006. Anysee: Peer-to-peer live streaming. In *Proceedings of IEEE Infocom.* 1–10.

REJAIE, R. AND STAFFORD, S. 2004. A framework for architecting peer-to-peer receiver-driven overlays. In *Proceedings of ACM NOSSDAV.* 42–47.

RIPEANU, M., FOSTER, I., AND IAMNITCHI, A. 2002. Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal 6,* 1 (September), 50–57.

SAROIU, GUMMADI, P. K., AND GRIBBLE, S. D. 2003. Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia systems 9,* 2 (August), 170–184.

SILVERSTON, T. AND FOURMAUX, O. 2006. P2p iptv measurement: A case study of tvants. In *Proceedings of International Conference on emerging Networking EXperiments and Technologies - CoNEXT.*

SILVERSTON, T. AND FOURMAUX, O. 2007. Measuring P2P IPTV systems. In *Proceedings of ACM NOSSDAV.*

STUTZBACH, D., REJAIE, R., AND SEN, S. 2008. Characterizing unstrucutured overlay topologies in modern p2p file-sharing systems. *IEEE/ACM Transactions on Networking 16,* 2 (April), 267–280.

SUH, K., FIGUEIREDO, D., KUROSE, J. F., , AND TOWSLEY, D. 2006. Characterizing and detecting skype-relayed traffic. In *Proceedings of IEEE Infocom.*

TRAN, D. A., HUA, K. A., AND DO, T. 2003. Zigzag: An efficient peer-to-peer scheme for media streaming. In *Proceedings of IEEE Infocom.* 1283–1292.

VU, L., GUPTA, I., LIANG, J., AND NAHRSTEDT, K. 2006. Mapping the pplive network: Studying the impacts of media streaming on p2p overlays. In *UIUC Tech report (UIUCDCS-R-2006-275).*

VU, L., GUPTA, I., LIANG, J., AND NAHRSTEDT, K. 2007. Measurement and modeling a large-scale overlay for multimedia streaming. In *Proceedings of IEEE QShine.*

WATTS, D. J. AND STROGATZ, S. H. 1998. Collective dynamics of small-world networks. *Nature 393*, 440–442.

WU, C., LI, B., AND ZHAO, S. 2007a. Characterizing peer-to-peer streaming flows. *IEEE Journal on Selected Areas in Communications, Special Issue on Advances in P2P Streaming Systems 25,* 9 (December), 1612–1626.

WU, C., LI, B., AND ZHAO, S. 2007b. Magellan: Charting large-scale peer-to-peer live streaming topologies. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS).* 62–69.

WU, C., LI, B., AND ZHAO, S. 2008. Exploring large-scale peer-to-peer live streaming topologies. *ACM Transactions on Multimedia Computing, Communications and Applications 4,* 3 (August).

XIE, S., KEUNG, G. Y., AND LI, B. 2007. A measurement of a large-scale peer-to-peer live video streaming system. In *Proceedings of International Conference on Parallel Processing Workshops (ICPPW).*

ZHANG, X., LIU, J., LI, B., AND YUM, T. S. P. 2005. Coolstreaming: A data-driven overlay network for peer-to-peer live media streaming. In *Proceedings of IEEE Infocom.* 2102–2111.