Computer Communications 35 (2012) 1004–1016

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Characterizing SopCast client behavior

Alex Borges [a,b,*], Pedro Gomes [a], José Nacif [a,c], Rodrigo Mantini [a], Jussara M. Almeida [a], Sérgio Campos [a]

[a] Computer Science Department, Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brazil
[b] Computer Science Department, Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais, Brazil
[c] Universidade Federal de Viçosa, Florestal, Minas Gerais, Brazil

ARTICLE INFO

Article history:
Received 24 February 2010
Received in revised form 15 June 2011
Accepted 28 February 2012
Available online 7 March 2012

Keywords:
Peer-to-peer
Live streaming
Client behavior
Internet measurement

ABSTRACT

Live streaming media applications are becoming more popular each day. Indeed, some important TV channels already broadcast their live content on the Internet. In such scenario, Peer-to-Peer (P2P) applications are very attractive as platforms to distribute live content to large client populations at low costs. A thorough understanding of how clients of such applications typically behave, particularly in terms of dynamic patterns, can provide useful insights into the design of more cost-effective and robust solutions.

With the goal of extending the current knowledge of how clients of live streaming applications typically behave, this paper provides a detailed characterization of clients of SopCast, a currently very popular P2P live streaming application. We have analyzed a series of SopCast transmissions collected using PlanetLab. These transmissions are categorized into two different types, namely, major *event* live transmissions and regular (or *non-event*) live transmissions. Our main contributions are: (a) a detailed model of client behavior in P2P live streaming applications, (b) the characterization of all model components for two different types of transmissions in the SopCast application, (c) the identification of qualitative and quantitative similarities and differences in the typical client behavior across different transmissions, and (d) the determination of parameter values for the proposed client behavior model to support the design of realistic synthetic workload generators.

1. Introduction

Streaming media applications are becoming more popular each day on the Internet. In fact, there are currently several video authoring resources, which make the creation and publishing of streaming media straightforward. This leads to a snowball effect: more content attracts more viewers, who, in turn, generate more content. Live media streaming, in particular, has attracted a growing number of users and broadcasters. For example, major television companies, such as NBC [1] and ESPN [2], distribute live content on the Internet, broadcasting their daily programs [3]. In addition, more and more home users are transmitting live content from their own desktops.

Live streaming media applications used to be based on the traditional client–server architecture. However, due to limited client scalability, a major weakness commonly associated with such centralized platforms, decentralized Peer-to-Peer (P2P) architectures are being increasingly adopted as platforms for live media distribution. In P2P systems, the clients (or peers) participate actively in content delivery, thus sharing with the server the total requirements for resources (e.g., processing and bandwidth capacities), and ultimately the total costs. In comparison with centralized architectures, in which the server sets up an independent (i.e., unicast) transmission to each client, the cost associated with each P2P live transmission imposed on the server is much lower. In other words, at a fixed total server cost, a P2P system is able to serve a larger number of simultaneous clients. Indeed, there are several currently very popular P2P live streaming applications. Some of them, such as PPLive and SopCast,[1] have reportedly reached the mark of 100 thousand simultaneous clients [4].

A large number of previous studies address the design of live streaming media protocols [5–9] and the structural organization of the P2P overlay network [10–12]. There are also some efforts towards characterizing the behavior of live streaming clients [13–17]. Client behavior patterns, which ultimately translate into system load patterns, directly impact system performance. Thus, a thorough understanding of them is key to drive future designs and optimizations [18,19]. However, previous analyses of P2P live streaming systems [13–17,20–23] address only a few aspects of client behavior (e.g., arrival process and client lifetime in the system), as several of them analyze the system from a different

* Corresponding author at. Computer Science Department, Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais, Brazil. Tel.: +55 32 3229 3311.
E-mail addresses: alex.borges@ufjf.edu.br (A. Borges), pcgomes@dcc.ufmg.br (P. Gomes), jnacif@dcc.ufmg.br (J. Nacif), mantini@dcc.ufmg.br (R. Mantini), jussara@dcc.ufmg.br (J.M. Almeida), scampos@dcc.ufmg.br (S. Campos).

[1] http://www.pplive.com and http://www.sopcast.com, respectively.

0140-3664/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.02.012

perspective (e.g., network traffic patterns [13,14] and quality of video transmissions [20]). In particular, we are not aware of any detailed characterization of client sessions, which indirectly reflect client dynamic behavior patterns (i.e., churn). As pointed out in [24], such patterns may have significant impact on P2P live streaming performance.

In this context, this article provides a detailed characterization of how clients of a popular P2P live streaming media application, namely SopCast, typically behave. Aiming at extending the knowledge provided by previous studies, our characterization addresses several aspects of client behavior, including client dynamic patterns. Our main goal is to provide data to support the future generation of realistic synthetic workloads, which, in turn, can be used to support both the evaluation of P2P live streaming protocols and the design of new P2P live applications.

Our characterization relies on a set of traffic logs collected from three different SopCast channels, using 421 crawlers in the PlanetLab network [25]. Two channels are Chinese TV channels delivering high bit-rate live news and sport content mostly to Chinese users. The third channel is a Brazilian entertainment channel transmitting live sport events at low bit-rate, mostly to Brazilian users. The collected traffic logs are used to reconstruct client behavior patterns during several transmissions of each channel. The monitored transmissions are categorized into two groups, based on the type of transmitted content: whereas the Brazilian channel transmissions broadcast decisive and important *live* events (i.e., the final games of a major soccer championship), the Chinese channel transmissions correspond to regular (i.e., *non-event*) live content (e.g., live TV news). In total, we characterize client behavior over 35 non-event transmissions and 2 event transmissions.

To drive our characterization, we propose a hierarchical model that captures the behavior of a client while watching a given channel transmission. At the higher *session level*, the model captures the multiple viewing sessions that the client may have of the given channel transmission. In other words, it captures the client dynamic behavior during the transmission. At the lower *partnership level*, the model captures the interactions the client establishes with one or more partners during a given session. We characterize a list of model parameters at each level including session interarrival times, number of sessions per client during a single transmission, session durations (or ON times), time intervals between consecutive sessions from the same client (or OFF times), number of partnerships, and partnership durations. Moreover, we analyze multiple transmissions of each type (event or non/event) separately, so as to account for possible variations across different transmissions.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 briefly reviews how a P2P live streaming system works and introduces the hierarchical model used to characterize client behavior in SopCast. Section 4 introduces the analyzed SopCast channels and our data collection methodology. Section 5 discusses temporal variations in the SopCast client population, whereas detailed characteristics of client behavior at the session and partnership levels are provided in Sections 6 and 7, respectively. A summary of our main findings along with a discussion of their implications for P2P live streaming systems are presented in Section 8. Section 9 concludes the paper and offers possible directions for future work.

2. Related work

A number of previous characterizations of live streaming media workloads are available in the literature, several of which analyze the workloads of popular P2P live applications, such as SopCast and PPlive [14–17,20–22]. However, these previous characterizations differ with respect to their goals, and thus to the perspective taken for workload analysis purposes.

Complementary to our work, some studies focus mainly on network traffic patterns, characterizing metrics such as upload and download rates, packet types, and protocol usage [13,14,17], while other efforts target the characterization of properties of the P2P overlay network, such as peer degree and clustering coefficient [16,21]. Silverston et al. [26] analyzed the level of fairness in the collaboration among peers, in terms of the ratio between upload and download traffic, as well as the geographic location of collaborating peers. One interesting conclusion they reach is that P2P-TV traffic typically crosses a large number of autonomous systems, which has important implications for Internet service providers. A few other studies analyze live transmissions from the perspective of the quality of the video received by the clients, characterizing metrics related to video continuity, visual quality and playback time differences among the participants [20].

Some previous studies also characterize metrics related to client behavior, such as geographical distribution, client lifetime in the system, client arrival rate, channel popularity and number of partnerships [13–17]. For instance, in [13,14], the authors analyze the transmissions of two live events (i.e., two soccer games of the 2006 World Cup) collected from 4 important applications, namely, PPLive, PPStream, TVAnts and SopCast, using 2 crawlers. Their main focus is on analyzing network traffic patterns. The only client behavior aspect analyzed is *lifetime*. Whereas the exact definition of lifetime is not very clear, it seems to encompass the time interval between the first and last packet sent or received by the client, thus possibly including multiple sessions (according to our definition) from the same client.[2] Thus, this metric cannot be directly compared to any of the metrics characterized here, as we analyze client behavior at the finer granularities of sessions and partnerships (within sessions).

In [17,22], the authors analyze the transmissions of 3 channels in the PPLive application. In addition to traffic related metrics, they also characterize some client behavior aspects such as peer geographical distribution, temporal variation of channel popularity, client arrivals and departures, lifetime, and number of partners per client. Like in [13,14], client lifetime is not directly translated into any of the metrics analyzed here. Moreover, the authors compare their results across 3 channels, exploring mainly channel popularity as a factor that may impact client behavior. In a complementary effort, we here analyze the impact of the type of transmitted content (event or non-event).

In [16], the authors analyze characteristics of the P2P overlay network in PPLive transmissions, comparing them against characteristics of existing file-sharing P2P overlays. The authors show that the network structure of P2P live transmissions can be well represented by a random graph, and that the average peer degree is independent of channel popularity. They also characterize client lifetimes, finding that, in comparison with file-sharing systems, live streaming clients tend to have shorter lifetimes. They perform the analyses separately for 3 different channels (e.g., a very popular channel, a channel transmitting a large set of short programs, etc.), although such channels are not characterized in terms of event and non-event transmissions.

In sum, previous studies of P2P live streaming workloads analyze only a few aspects of client behavior. In particular, none of them analyze the properties of client sessions, thus neglecting that a client may join and leave the same transmission multiple times, which may ultimately affect the system dynamics and performance. Moreover, none of them compare client behavior across

[2] We here use the term *lifetime* to refer to such period so as to make it clear that it is *different* from our definition of session ON time. However, we should note that some previous studies refer to such period as a *session*.

transmissions that differ, fundamentally, in terms of the type of content, e.g., an important live event transmission versus a regular (non-event) live transmission.

Thus, to the best of our knowledge, this work is the first that focuses primarily on client behavior, capturing several aspects that jointly show a clearer and more detailed picture of how clients interact with SopCast, and addressing potential qualitative and quantitative differences that may exist between major event transmissions and regular non-event transmissions. Our characterization results provide valuable insights to support the future design of more realistic synthetic workload generators as well as of new protocols that exploit client behavior patterns, taking the type of transmitted content into account.

3. P2P Live streaming

In this section, we first briefly overview the main components of a P2P live streaming system (Section 3.1). Next, we introduce the client behavior model that drives our characterization of the SopCast system (Section 3.2).

3.1. System components

P2P live streaming systems are composed of clients (peers or nodes) that collaborate to disseminate the media content. The clients are organized into a virtual overlay network, on top of the real computer network. Each channel, while transmitting a live stream, has its own P2P overlay network, independent from any other channel maintained by the application. In other words, each live channel is an autonomic P2P network. Thus, we use the terms "transmission", referring to an active channel transmitting live content, and "P2P network" interchangeably throughout this paper.

The P2P overlay network is commonly based on either a structured tree-like overlay or a non-structured mesh-based overlay. In both architectures, there is a special client called server S, which generates the live content, splitting the media into pieces called chunks, which are transmitted over the P2P network for later exhibition. In order to participate, clients receive the streaming media and relay the content to its partners.

Most currently popular P2P live streaming applications, such as SopCast, PPLive and GridMedia,[3] use a non-structured mesh-based overlay network, known as data-driven mesh-based overlay network [27]. In this overlay architecture, a client explicitly requests a needed media chunk from one of its partners. It tends to be resilient to failures and client dynamics, providing a smooth streaming media visualization. This architecture, adopted by SopCast, drives our client behavior model, introduced in the next section.

To join the system, a client c_i first registers itself at a centralized bootstrap server B, which normally is distinct and independent of the overlay network. B then returns to c_i a subset of the currently active clients, which is a list of potential partners of c_i, LPC_i. Thus, given C the set of active clients (according to B) in the system at the time that c_i contacts B, $LPC_i \subseteq C$ and $LPC_i \neq \emptyset$. After this start-up mechanism, the joining client c_i then selects n clients from LPC_i and tries to establish partnerships with each of them. Successfully established partnerships determine LP_i ($LP_i \subseteq LPC_i$), the set of partners of c_i.

While in the system, c_i periodically exchanges keep-alive messages with its partners. When c_i detects that a partner c_j ($c_j \in LP_i$) is idle (i.e., c_i stops receiving keep-alive messages from c_j), c_i removes c_j from LPC_i and LP_i, selects another client from LPC_i, and tries to establish a new partnership with it. It may also choose to contact the bootstrap server to get a new list of potential partners. Thus, sets LPC_i and LP_i are dynamically updated.

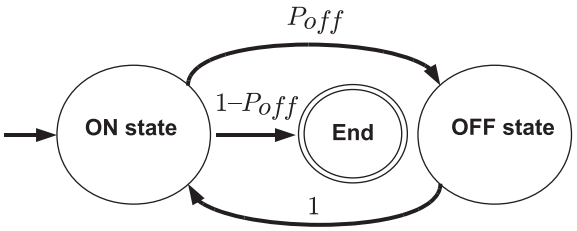


Fig. 1. P2P live streaming: client ON/OFF model.

Active clients exchange media chunks only with their partners. A client c_i periodically checks which chunks are needed, identifies which partners can have these chunks, and sends requests for them. More precisely, each client c_i maintains a chunk map cm_i, which specifies the chunks that c_i currently has in its streaming buffer (and thus can be forwarded to other clients) as well as the chunks it still needs. If c_i does not have a chunk h, it marks $cm_i[h] = 0$; otherwise, $cm_i[h] = 1$. Clients often exchange their chunk maps with their partners, thus learning each other's needs and chunk availabilities. During a channel transmission, a client sends requests for data chunks to their partners or directly to the server S as well as serve requests from their partners by forwarding previously received chunks to them [27].

3.2. Client behavior model

Our proposed client behavior model assumes that each client c_i can join and leave the P2P network (i.e., the live transmission) at any time. In other words, we assume an ON/OFF model, in which each client alternates between an active – *ON* – state, and an idle – *OFF* - state. While in the *ON* state, client c_i is in a *session*, exchanging media chunks with its partners, trying to fetch and rebuild the original streaming data for exhibition on a media player. When c_i leaves the *ON* state, it may go to an *OFF* state with probability P_{off}, or it may quit the transmission, never returning, with probability $1 - P_{off}$. Either event marks the end of the current session of c_i.

While in the *OFF* state, c_i does not exchange any media chunk with its partners, remaining completely inactive. After a while, the client rejoins the P2P network, starting a new *session*, during which it will exchange more chunks and interact with other clients. This dynamics of client behavior is captured by the simple state machine shown in Fig. 1.[4]

Based on this ON/OFF model, we propose a hierarchical model that captures the behavior of a client c_i in terms of its interactions with the system, that is, its behavior during *a single live channel transmission*. Our model, illustrated in Fig. 2, captures client behavior at two different levels. At the higher *session level*, the model captures the multiple viewing sessions that c_i may have of the given channel transmission. For instance, in Fig. 2(a), c_i joins the live transmission, has two viewing sessions during that transmission, and then leaves for good. At the lower *partnership level*, the model captures the interactions c_i establishes with one or more partners, *during a given session*. In other words, each client session is composed of potentially multiple partnerships. In Fig. 2(b), c_i establishes three partnerships during its first session. Our hierarchical approach is similar, in nature, to the approaches adopted by previous characterizations of other types of workloads [28,29].

Given this hierarchical model, client behavior for a given live transmission can be characterized according to the following set of parameters (or metrics). First, we need to define the rate at

[3] http://www.gridmedia.com.cn.

[4] Note that, by definition, an *OFF* state occurs between two consecutive *ON* states. Thus, the *END* state captures the final departure of the client from the P2P network.

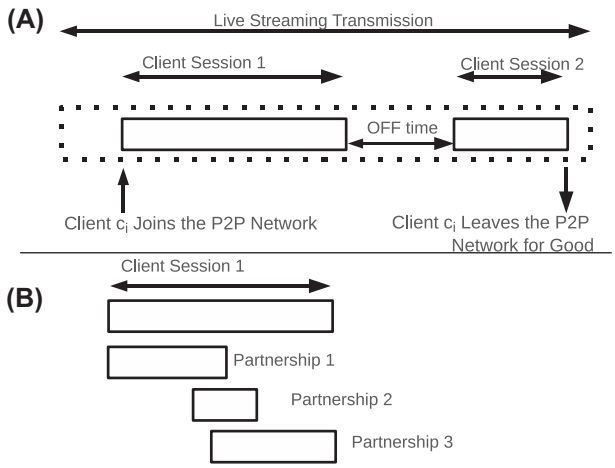


Fig. 2. P2P live streaming: hierarchical client behavior model.

which new client sessions are initiated. Thus, we define the *session inter-arrival time* as the time elapsed between the beginning of two consecutive sessions, regardless of whether they are from the same client or from different clients. We also need to define the *number of sessions* that each client c_i has during the same transmission.[5]

We define *ON time* as the period of time during which a client is active in the transmission, that is, exchanging streaming data with any of its partners. Thus, an ON time is equivalent to the duration of a client session, and is defined as the period of time elapsed between the first and the last *data* packet exchanged between the client and any of its partners during the corresponding session. We also define *OFF time* as the time interval between the end of a session and the beginning of the next session of the same client (during the same transmission).

During each session, a client establishes one or more partnerships. Thus, at the partnership level, client behavior can be characterized in terms of *number of partnerships* established within a single session, and *partnership duration*. We focus on what we call *active partnerships*, which refer to pairs of clients exchanging *data* packets, as our goal is to analyze the behavior of clients viewing the live content. We do not consider trading control packets as part of an active partnership. Thus, the duration of a partnership between two clients is defined as the time elapsed between the first and the last data chunk exchanged between them, within the current session. Note that, during a single session, c_i establishes at most one partnership with another given client c_j, although c_i may establish multiple partnerships, each one with a different client. Moreover, the duration of a partnership is upper-bounded by the duration of the session (i.e., ON time) during which it was established.

4. SopCast data collection

Our client behavior characterization is based on traffic logs collected from SopCast, one of the most popular P2P live streaming applications.[6] SopCast currently offers a great variety of channel options. In Section 4.1, we present the channels that we monitored, whereas our crawling methodology is described in Section 4.2. Section 4.3 briefly overviews the collected data.

4.1. Analyzed channels

The three analyzed SopCast channels were selected because of their different characteristics in terms of country of origin, target audience, coding, image quality, and program content. The first channel is CCTV, a state-owned Chinese news broadcaster. It is very popular in China, and transmits higher quality video in comparison with the other two channels (around 600 kbps). The second channel is a sports specialized channel in Chinese language. Usual program examples include sport news, live game transmissions and sport documentaries. It also transmits reasonably high quality content (around 500 kbps). The last channel transmits the content of the most popular TV station in Brazil. It was monitored during two major sport events for the Brazilian audience, i.e., the final games of an important soccer championship. During these transmissions, the live content was broadcasted at medium to low rates for SopCast standards (around 250 kbps).

As will be discussed in the following section, we have collected a number of traffic traces during different transmissions of each channel. The monitored transmissions can be grouped into two categories, in terms of the type of content: (1) *event* transmissions refer to the live broadcast of some major and decisive event of any nature (e.g., sports, political, etc.), whereas (2) *non-event* transmissions broadcast regular live content. The transmissions in the Brazilian channel fall into the *event* category, whereas the transmissions in the two Chinese channels correspond to non-event transmissions.

4.2. Crawling methodology

Modeling client behavior in P2P live streaming systems (and SopCast in particular) is a challenging task [17]. The information stored in the SopCast tracking servers is not available for public use, thus making it difficult to produce a precise reconstruction of the P2P network characteristics. In order to overcome this limitation, we applied a methodology to collect data and to reconstruct the P2P network that is similar to the one used in [16,20].

In particular, we set up a group of computers running the SopCast application as well as network monitoring tools. These data crawlers joined the SopCast network as ordinary clients, and actively recorded interactions performed between them and other peers. After a period of time, all the data collected by these computers were merged into a single trace, which was used to reconstruct the SopCast network dynamics during the crawling period.

More precisely, our methodology is composed of two main phases, namely, (1) setup configuration, and (2) data crawling. Setup configuration starts by letting all data crawlers update their versions of the local softwares, including Wireshark[7] (tcpdump), which is used to capture all network traffic observed during the second phase. Next, following [30], the crawlers synchronize their local times according to a given server, using Network Time Protocol[8] – NTP [31]. We used the standard configuration of the PlanetLab nodes for NTP servers.[9]

During the second phase, all data crawlers first join the same (target) SopCast channel. Joining times are normally distributed over a given initial period T. In our analyses, we only consider data collected after *all* crawlers are logged in SopCast, disregarding any data received during the initial period T. Thus, any congestion and overloading that might occur during this initial joining period, which might affect client behavior, is not included in the analyzed

[5] Note that this number is related to probability P_{off}.

[6] We selected SopCast because: (1) according to Google Trends (http://www.google.com/trends), it receives a larger number of searches than other popular applications such as PPLive and PPStream; and (2) a Linux implementation of SopCast is available, which enables our experiments on PlanetLab.

[7] http://www.wireshark.org.

[8] This makes the local time differences among data crawlers quite small (less than 1 s). Given that our client behavior characterization considers a time granularity of 1 s, such differences can be considered negligible, for practical terms.

[9] If this configuration was not available, we used public NTP servers such as ntp.ubuntu.com.

Table 1
Ratio between the numbers of simultaneous clients discovered in each network snapshot using k crawlers and using all 421 crawlers (averages and SEM values, as fractions of corresponding averages, in parenthesis).

$k = 2$	$k = 10$	$k = 50$	$k = 70$	$k = 100$	$k = 200$
13.5% (0.11%)	42.1% (0.07%)	85.5% (0.05%)	88.9% (0.04%)	93% (0.08%)	98% (0.1%)

data. As soon as it joins the channel, each data crawler uses Wireshark to capture all network packets with origin and destination port numbers equal to those configured for the SopCast application (i.e., in-bound port number equal to 8901, out-bound port number equal to 8902). In preliminary experiments, we also monitored other ports to check whether SopCast uses additional ports. Since we found that no other port was used, we opted to restrict our crawling procedure to the configured port numbers.

The data captured by each crawler consists of the timestamp (at 1-s granularity) of each packet sent (received) to (from) other SopCast clients and packet size information. After crawling finishes, the data captured by all crawlers are sent to an application developed by us. Based on the collected packet size and timing information, this application reconstructs the behavior of the SopCast clients that had some contact with the crawlers during the crawling phase. That is, the SopCast P2P network dynamics is reconstructed as a sequence of network snapshots, taken at 1 s intervals.

We used 421 PlanetLab computers as SopCast crawlers. We selected computers located in various parts of the world in order to achieve a large coverage of the SopCast network. We did not impose any restriction on each computer's upload and download capacity. We note that all 421 crawlers remained active in the system (no churn) throughout data collection. Although such a large number of stable nodes might introduce some bias in our measurements, we conjecture that such bias, if present, might not be very significant because, as we shall see, the same general client behavior patterns were observed across all transmissions of the same type (e.g., all non-event transmissions), despite the large variability in the number of simultaneous clients connected to such transmissions. In particular, for some of those transmissions, the crawlers represented a small fraction of all peers. Thus, we conjecture that our crawlers worked mostly as network observers, and did not greatly impact how real clients behave.

We also note that, in comparison with previous P2P live streaming characterizations [13,14,16,17,20,32], we use a larger set of crawlers (previous efforts used at most 70 crawlers [20]). To assess the benefit of using such a larger number of crawlers, we analyzed the impact of the number of crawlers on the number of *simultaneous* clients discovered by them on each *snapshot* of the network (i.e., during each 1 s interval). That is, we analyzed the number of simultaneous clients discovered by a subset of k crawlers, randomly selected among our 421 crawlers, for various values of k. For each such value, we repeated the random selection 200 times, and considered each 1 s snapshot of the P2P network in all analyzed transmissions.

Table 1 shows average results for the ratio between the number of simultaneous clients discovered by k crawlers and the number of simultaneous clients discovered by all 421 crawlers. It also shows, in parenthesis, the standard deviations of the averages, also known as Standard Error of the (sample) Mean (SEM) [33], as measures of the variability of the results.[10] SEM values, shown as fractions of the corresponding averages, are typically small, implying a small variability across different snapshots and transmissions.

As shown in the table, using at most 70 clients (as opposed to 421) leads to a loss of at least 11%, on average, in the number of simultaneous clients. However, there are clear diminishing returns: the benefit of having more than 200 crawlers is only marginal. As we had no way of knowing these results beforehand,[11] we used the largest number of stable PlanetLab nodes that we could gather during our experiments, that is, 421 nodes. Although this number seems larger than it is necessary, it does bring benefits over using only at most 70 crawlers, as the client population is larger and possibly more representative of the complete network. Moreover, the observed diminishing returns effect provides evidence (though no guarantee) that we might have collected the vast majority of all clients.

4.3. Overview of our data collection

We collected SopCast data from December 2008 to January 2009. We performed two crawlings of the Chinese channels. First, we monitored both channels for a whole week in order to analyze temporal variations in channel popularity (i.e., number of simultaneous clients). The data collected during this first crawling is analyzed in Section 5.

We then selected one hour of observed peak popularity – 8PM local time – and monitored each Chinese channel for 100 min, during several days. We chose 100-min long transmissions as this was the (approximate) duration of the two analyzed event transmissions. In total, this second crawling produced 35 different data sets. These 35 non-event transmissions, together with 2 event transmissions monitored from the Brazilian channel, are used to analyze session and partnership characteristics in Sections 6 and 7.

Each analyzed transmission has between 4385 and 12,233 client sessions. Event transmissions served up to 3000 simultaneous clients, whereas non-event transmissions had from 700 to up to 6000 simultaneous clients. These numbers represent instantaneous measures computed over each network snapshot (i.e., 1 s time interval). Moreover, for each transmission, each crawler collected data related to over 2,000,000 interactions and more than 3000 unique partners, where each interaction corresponds to one (data or control) packet sent or received by the crawler. In each snapshot of the network, each crawler had around 90 simultaneous partners. These numbers are larger than those reported in previous analyses. In [16], for instance, the authors report that the analyzed PPLive channels had at most 2500 simultaneous clients, and crawlers had about 35 simultaneous partners.

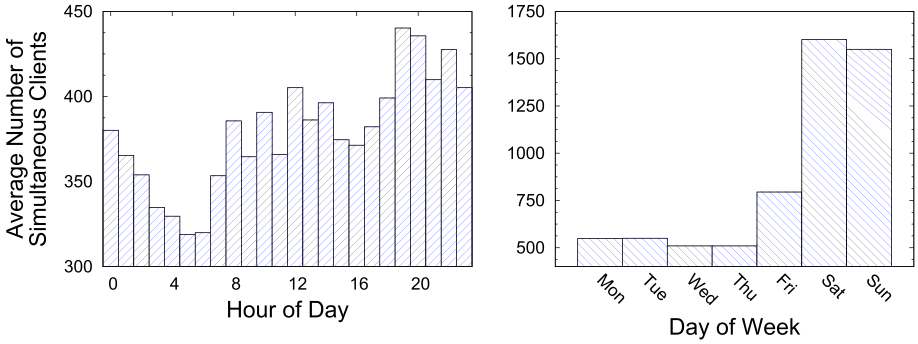
5. Temporal variations in channel popularity

We start our characterization of SopCast client behavior by analyzing hourly and daily variations in channel popularity, that is, in the number of simultaneous clients connected to a given channel. We here focus on the two Chinese channels (non-event transmissions), as the monitored event transmissions last for only 100 min.

Fig. 3 shows representative results for the CCTV channel during one specific weekday, namely, December 16th, 2008 (Fig. 3(a)), and one specific week (Fig. 3(b)). The hourly variation pattern is typical of what has been observed in several weekdays for the two monitored channels. The number of simultaneous clients starts growing at 7 AM local time, remaining roughly stable until around 4 PM. Channel popularity starts increasing once again, reaching a peak around 8PM and staying roughly stable for about 3 h. Thus, the

[10] The Standard Error of the (sample) Mean, SEM, of a sample of n observations is computed as the ratio between the standard deviation of the sample s and the square root of the number of samples n, i.e., $\frac{s}{\sqrt{n}}$. Note that a $(1-\alpha)$% confidence interval for the sample average can be determined by multiplying the corresponding SEM value by the $(1-\alpha/2)$-quantile of the z or t variate, depending on the number n of observations [33].

[11] Note that previous analyses similar, in nature, to ours [32] considered only the total client population discovered during the whole transmission as opposed to the client population simultaneously connected to the system (as we do here).



a) Hourly Variation. b) Daily Variation.

Fig. 3. Temporal variations in client population in the CCTV (non-event) channel.

period with the largest number of simultaneous clients seem to coincide with the time when people arrive at home, after work or school.

As shown in Fig. 3(b), non-event SopCast channels present daily variations in the number of simultaneous clients, with significant peaks around the weekend, similarly to what was observed in [21]. On Saturday and Sunday, the channel receives, on average, as many as 3 times more simultaneous clients than on Monday through Thursday. The much higher popularity on weekends may be explained by a larger number of users who spend their spare time searching for some entertainment on the live channels.

In the next sections, we analyze sessions and partnerships of clients of non-event transmissions during periods of peak channel popularity. We also considered periods of rough stability, in terms of number of simultaneous clients, to avoid impacting the analyzed distributions with diurnal patterns [28]. Thus, based on the results shown in Fig. 3(a), we chose to monitor the non-event channels starting at 8PM, for 100 min, across several days.

6. Session level characterization

In this section we characterize each session level component of the hierarchical client behavior model presented in Section 3.2. We start by first discussing how we identify client sessions (Section 6.1). Next, we analyze session inter-arrival times (Section 6.2), number of sessions per client (Section 6.3) as well as ON and OFF times (Sections 6.4 and 6.5). Since the PlanetLab crawlers remained active in the system throughout the collection period, their behavior, in terms of session dynamics, does not necessarily reflect the behavior of real SopCast clients. Thus, we disregard them from our analyses of client session characteristics, focusing, instead, on real SopCast clients that have interacted with any of our crawlers during data collection.

We note that, for each model component analyzed in this section as well as in Section 7, we quantify how close the distributions of measured data found for different transmissions are to each other. We do so by computing the average and corresponding SEM value for each distribution percentile.[12] As we will see, SEM values tend to be small if computed for transmissions of the same type (event or non-event), meaning that the same distribution fits reasonably well the data measured across all transmissions of a given type.

Moreover, we characterize such distributions by presenting statistical models that best fit the measured data. The best fitted distribution is defined by comparing the least square errors (LSE) [33] of the best fitted curves for a number of commonly used distribution models. The following distribution models are considered as candidates for best fit: Exponential, LogNormal, Weibull, Pareto, Gamma and Normal.[13] We also visually compare the curve fittings both at the body (small values in the x-axis) and at the tail (large values in the x-axis) of the measured data to support our fitting decisions.

6.1. Identifying client sessions

SopCast does not explicitly delimit client sessions. If a client stops interacting with its partners, one cannot precisely determine if it has left the system or if it is experiencing temporary network problems. In particular, although the protocol does have control packets to indicate the beginning of a partnership (as discussed in [34]), there is no control packet marking the end of it. Moreover, we observed, in some preliminary experiments, that a client may continue receiving control packets from its partners for more than 15 min after leaving the SopCast network. Thus, we are not able to precisely delimit a session by simply inspecting control packets in our logs.

Therefore, we follow an alternative approach that was adopted by several previous studies to delimit client sessions in various types of applications [28,29,35]. That is, we assume that a session ends when the client remains inactive for a period of time exceeding a certain threshold. Periods of inactivity below the given threshold account for delays caused by temporary connection break downs or network traffic congestion, which may happen in real networks, and should not necessarily cause session interruption.[14]

To determine the session threshold, we observe the distribution of time intervals between two consecutive data packets exchanged between a given client and any of its PlanetLab partners. More precisely, we define the *inter-activity time* as the period of time elapsed since a given client sends/receives a data packet to/from any of its PlanetLab partners until it sends/receives another data packet (also to/from any of its PlanetLab partners). That is, it is the time period between two consecutive activities (data reception or transmission) of the client, as perceived by the PlanetLab crawlers. We then search

[12] That is, let us assume we have n data sets, collected during n different transmissions. We first take the sample distribution for each individual transmission. Next, for each distribution percentile i ($i = 1 \cdots 100$), we compute the average $\overline{x_i}$ and the standard deviation s_i across all n distributions. Finally, we compute the standard deviation of the mean $\frac{s_i}{\sqrt{n}}$. This corresponds to the SEM value for percentile i. We do so for various percentiles [33].

[13] Functions (PDF) of these distributions are: Weibull: $p_X(x) = \alpha\beta x^{\beta-0} e^{-\alpha x^\beta} I_{(0,\infty)}(x)$, LogNormal: $p_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\frac{-(\ln(x)-\mu)^2}{2\sigma^2}}$, Exponential: $p_X(x) = \lambda e^{-\lambda x}$, Pareto: $p_X(x) = \frac{\alpha k^\alpha}{x^{\alpha+1}}$, where $x \geqslant k$, Gamma: $\Gamma(a) = \int_0^\infty s^{a-1} e^{-s} ds$, and Normal: $P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 2\sigma^2}$

[14] Note that, by definition, OFF times cannot be shorter than this threshold.

for a significant change (e.g., a knee) in the shape of the distribution of inter-activity times, which might reflect a change in the underlying process, possibly delimiting different regimes and indicating the aggregation of multiple distributions (e.g., short times could indicate regular idle times due to network-related delays, whereas longer times reflect client inactive behavior).

In order to plot the distribution of inter-activity times, we first need to identify data packets. To do so, we rely on previous studies of SopCast and PPlive [13,14,17] which showed that packet size is a good heuristic to determine if a network packet corresponds to a data or control packet. According to them, streaming data packets tend to be larger, around 1100 bytes, whereas smaller packets commonly represent control packets. Indeed, we did observe, in some preliminary experiments, that SopCast control packets have around 70 bytes. Thus, for analysis purposes, we consider as a data packet any network packet carrying more than 1000 bytes.

Having identified the data packets, we now turn to the analysis of the distribution of inter-activity times. We observed that all 37 event and non-event transmissions present very similar distributions. Indeed, the SEM values, computed for each percentile across all 37 individual distributions, are below 0.6% of the average values. Thus, for the sake of presentation, we aggregate all 37 distributions into a single "average" distribution, obtained, by taking, for each percentile, the average across all distributions.

We start by noting that the vast majority (89%) of the measured inter-activity times are very short, i.e., under 5 s. Such short time intervals are most likely due to network-related delays. Thus, a plot of the complete distribution of inter-activity times would be dominated by these very short time intervals, making it hard to determine the session threshold, since the relevant portion of the curve is obfuscated. Thus, in order to be able to analyze the distribution of inter-activity times in search for a meaningful threshold, we focus on the relevant part of the curve, removing all measured data points that are under 5 s. The cumulative distribution computed over the remaining data points is shown in Fig. 4. Small bars along the curve indicate the SEM values measured at different percentiles.

Note that the cumulative probability grows fast until 120 s. After that, the growth slows down, and the curve presents a knee around 150 seconds. Based on this observation, we choose 150 s as the session threshold value. We also experimented with other values between 120 and 180 s, finding no significant difference in the characterization results.

6.2. Session inter-arrival times

Fig. 5 shows the complementary cumulative distributions of session inter-arrival times for event and non-event transmissions. We found that the individual distributions computed for each of the 35 non-event transmissions are very similar to each other (i.e., very small SEM values). Similarly, the distributions for both event transmissions are also very close to each other, though very different from the distribution for non-event transmissions. Thus, for the sake of presentation, Fig. 5 shows a single "aggregated" distribution for each transmission type. Each distribution was obtained by computing the average percentiles across the distributions for all transmissions of the same type. The figure also shows SEM values for various distribution percentiles. The SEM values are under 0.5% of the corresponding averages, for all percentiles of both aggregated distributions.

Session inter-arrival times tend to be much shorter during event transmissions. The monitored events attracted a large number of clients, most of them arriving at the beginning of the transmission. We note that the fraction of session inter-arrival times greater than 10 s is only 0.3%. In contrast, for non-event transmissions, this fraction grows to 3%, which is not negligible. Nevertheless, for both types of transmissions, the majority of session inter-arrival times are quite short, under 2 s, indicating high session arrival rates. Indeed, event and non-event transmissions receive, on average, 0.705 and 0.495 sessions per second, respectively.

In order to determine the statistical models that more closely fit the distributions of session inter-arrival times, we compared the LSE values of the best fitted curves for the several considered alternatives. We also visually compared the curve fittings at the body and at the tail of the measured data. We found that LogNormal distributions, with different parameters, are the best fits for both event and non-event transmissions. Fig. 5 shows the two fitted LogNormal distributions, whereas Table 2 summarizes our findings providing the mean, standard deviation, best-fitted distribution model and distribution parameters for each type of transmission.

Note that, for both event and non-event transmissions, the LogNormal distributions overestimate the probabilities of very short inter-arrival times. Yet, among all tested models, they are the ones that more closely approximate the data, even if we consider only this portion of the curves. Though not ideal, overestimating those probabilities is preferable to underestimating them, since short inter-arrival times (e.g., burst arrivals) have a stronger impact on server capacity planning and content sharing. Thus, overestimating their probabilities lead to more conservative system design decisions, which may be preferable to more aggressive, and possibly riskier, choices.

6.3. Number of sessions per client

We now characterize the number of sessions that each client has during one transmission. A larger number of sessions imply that the client leaves and (re) joins the channel multiple times during the same transmission.

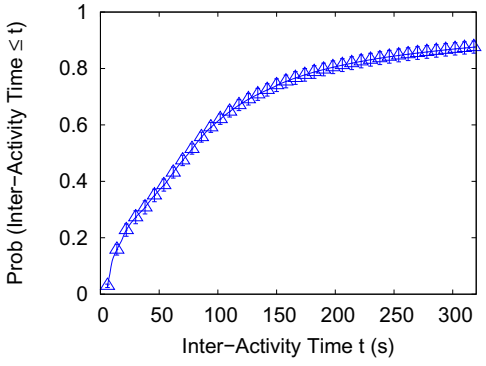


Fig. 4. Inter-activity times for event/non-event transmissions (computed over observations greater than 5 s).

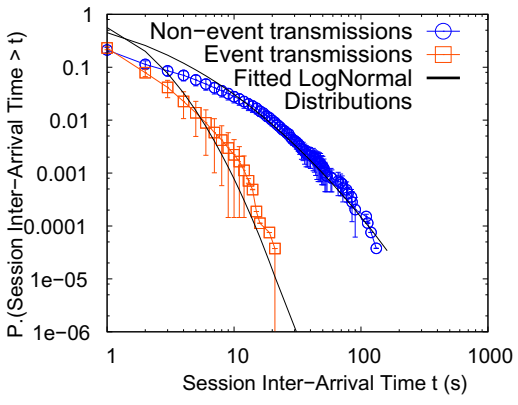


Fig. 5. Session inter-arrival times.

Table 2
Distributions of session inter-arrival times: summary.

Transmission type	Best fit	Mean (s)	Std. Dev. (s)	First parameter	Second parameter
Non-event	LogNormal	2.012	4.371	$m = -0.166$	$\sigma = 1.318$
Event	LogNormal	1.417	1.113	$m = 0.108$	$\sigma = 0.693$

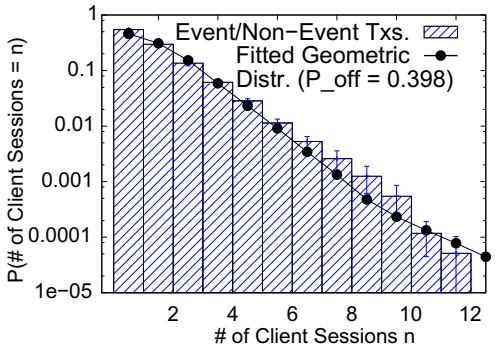


Fig. 6. Number of sessions per client during a single (event/non-event) transmission.

We found very similar distributions of the number of sessions per client in all 37 (event and non-event) transmissions. Indeed, the SEM values computed over all 37 distributions are under 0.6% of the corresponding averages. Thus, Fig. 6 shows a single complementary cumulative distribution, computed by averaging percentiles across all 37 transmissions. The figure shows that most clients have only a few sessions during one transmission. Indeed, around 54% of the clients leave the system after a single session, whereas 80% of them have at most 2 sessions per transmission.

Recall that, considering the ON/OFF model on which our hierarchical client behavior model is built, the number of sessions per client can be computed based on the transition probability to state *OFF*, P_{off}. In other words, it follows a Geometric distribution[15] with parameter $1 - P_{off}$. Fig. 6 shows the Geometric distribution that best fitted our measured data, for which $P_{off} = 0.398$. Note the good agreement between both curves.

6.4. Session ON times

We now analyze session ON times, expressing them as percentages of the total transmission duration. We chose to characterize *normalized ON times*, as opposed to absolute measures, as the fitted distributions can be more directly applied into generating synthetic workloads, with no need of data transformations to deal with boundary conditions (e.g., synthetically generated ON times that exceed the transmission duration). We note that, for the specific transmissions analyzed in this paper, normalized ON times correspond (approximately) to absolute measures taken in minutes, as each monitored transmission lasts for (approximately) 100 min.

Once again, we found that the distributions of session ON times follow similar patterns for most transmissions of the same type, particularly for non-event transmissions. As a matter of fact, the SEM values computed over the 35 individual distributions obtained for the non-event transmissions do not exceed 0.72% of the reported average values. For the two event transmissions, SEM values are somewhat larger, possibly due to the smaller sample size. Nevertheless, they are still reasonably small, falling under 5.7% of the averages.

[15] The probability mass function of a Geometric Distribution with parameter p is given by $Prob(x = k) = (1 - p)^{k-1}p$.

However, we did find very distinct distributions for event and non-event transmissions, as shown in Fig. 7. The (aggregated) distribution of ON times for event transmissions (Fig. 7b) is much more skewed towards larger numbers, meaning that session ON times tend to be much longer in such transmissions. While for regular non-event transmissions (Fig. 7(a)), 10% of the client sessions last for at least 20% of the transmission, this fraction grows to 33% for event transmissions. Indeed, a considerable number of clients (around 15%) stay online throughout most of the transmission (ON times greater than 90%), which may be expected for transmissions of the final games of a major soccer championship.

As shown in Fig. 7(a), for non-event transmissions, Gamma and LogNormal distributions provide the best fits for the measured ON times. Whereas the LogNormal distribution is a better fit for smaller values, which account for a large fraction of all measured data, Gamma more closely approximates the tail of the curve. In contrast, Fig. 7(b) shows that ON times in event transmissions are better approximated by a Weibull distribution. These results are summarized in Table 3.

6.5. Session OFF times

In this section, we characterize the final component of our session level client behavior model, namely, OFF times. Like with ON times, we analyze *normalized OFF times*, expressed as fractions of the transmission duration.

We found that, unlike ON times, OFF times follow very similar distributions across all 37 transmissions, regardless of type (event or non-event). Indeed, the SEM values computed over all 37 distributions are below 2.5% of the averages. Moreover, as shown in Fig. 8, OFF times are typically small: around 80% of the measured samples are shorter than 30 min, while 40% of them are shorter than 10 min. Thus, although many clients have only one session during a transmission, a considerable fraction of the clients that have multiple sessions (and thus experience OFF times) tend to return to the transmissions within somewhat short time intervals. This finding may be exploited in the design of partnership policies that take into account the probability that an inactive client returns to the system shortly.

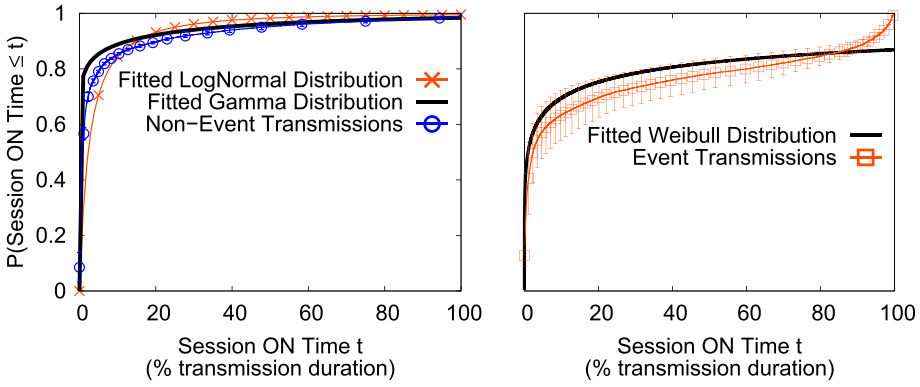
We also found that OFF times are well fitted by an Exponential distribution, both at the body and at the tail of the curve. Indeed, Fig. 8 shows a very good agreement between measured data and fitted distribution. A summary of our characterization of OFF times is presented in Table 4.

7. Partnership level characterization

We now turn to our characterization of the partnership level components of our proposed client behavior model. In particular, we characterize the number of partnerships established by each client within a single session (Section 7.1) and the durations of such partnerships (Section 7.2).

Recall that we here focus on *active partnerships*, which correspond to periods of time during which two clients exchange chunks of streaming *data*. Moreover, we here analyze the partnerships established between each PlanetLab crawler and other clients (including real clients and other crawlers). In other words, we do not analyze partnerships established between pairs of real SopCast clients (i.e., non-crawlers), as we are not able to precisely monitor all partnerships established by them.[16]

[16] As previously mentioned, we noticed, in preliminary experiments, that a client may remain in the list of partners of another client for more than 15 min after it has left the system. Thus, monitoring a client's list of partners is prone to overestimations.



a) Non-Event Transmissions. b) Event Transmissions.

Fig. 7. Normalized ON times.

Table 3
Distributions of normalized ON times: summary.

Transmission type	Best fit	Mean (%)	Std. Dev. (%)	First parameter	Second parameter
Non-event	LogNormal	6.60	17.96	$m = 0.823$	$\sigma = 1.459$
	Gamma			$\alpha = 0.062$	$\beta = 107.3$
Event	Weibull	23.59	34.99	$\alpha = 2.032$	$\beta = 0.233$

Table 4
Distribution of normalized OFF times: summary.

Transmission type	Best fit	Mean (%)	Std. Dev. (%)	Parameter
Event/non-event	Exponential	18.49	16.17	$\lambda = 0.054$

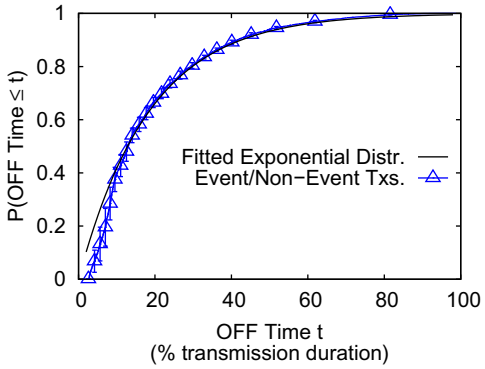


Fig. 8. Normalized OFF times.

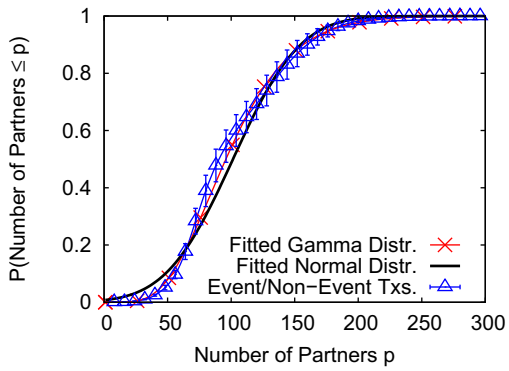


Fig. 9. Number of Partnerships per client (measured at each snapshot of the network).

7.1. Number of partnerships

Note that, by definition, a given client may have at most one partnership with any other participant of the system, since this partnership is analyzed within the context of a single session. Thus, we analyze the number of unique partners each PlanetLab crawler has during each collected network snapshot.

We found very similar distributions of the number of partnerships per client across all 37 transmissions, regardless of type. Indeed, SEM values computed over all 37 individual distributions are under 1.67% of the averages. This might be explained, at least partially, by internal mechanisms of the application, which seem to give incentives for clients to keep a number of partners within certain bounds or to periodically choose new partners.

Fig. 9 shows the (aggregated) distribution as well as SEM values for various percentiles. On average, a client keeps around 101 simultaneous partners (i.e., during any single snapshot). This result is in close agreement with results from a previous characterization of 4 different applications, including SopCast [13]. Moreover, around half of the clients have at least 90 partners in any snapshot of the system, which might indicate a greedy neighborhood policy implemented by SopCast.

As shown in Fig. 9, both Gamma and Normal distributions provide good fits for the distribution of number of partnerships. Although the Gamma distribution fits somewhat better the measured data (i.e., with a smaller least square error), the Normal distribution, which is simpler to compute, does capture quite reasonably the measured data, and thus may also be used to generate realistic synthetic workloads with no significant difference. These findings are summarized in Table 5.

As a final note, we also analyze the similarity between the lists of partners of different clients at any given time. In other words, we analyze the clients that appear *at the same time* (at the 1-s granularity) in the lists of partners of groups of PlanetLab crawlers, computing the overlap between them. We do so by computing, for a given snapshot, the ratio between the number of partners shared by all crawlers in the group and the total number of unique partners in the union of their lists. We repeat this procedure for various snapshots of several transmissions, computing the ratio for various group sizes, selected according to one of the following criteria: (1)

Table 5
Distribution of number of partnerships per client: summary.

Transmission type	Best fit	Mean (# partners)	Std. Dev. (# partners)	First parameter	Second parameter
Event/non-event	Gamma	101.453	41.537	$\alpha = 6.008$	$\beta = 16.886$
	Normal			$\mu = 101.453$	$\sigma = 41.537$

Table 6
Overlap in the list of partners of groups of crawlers (averages and SEM values, as fractions of corresponding averages, in parenthesis).

Type of group	Number of crawlers in the group			
	2	3	4	5
Randomly selected	6.576% (0.098%)	1.27% (0.001%)	0.328% (0.22%)	0.133% (0.313%)
Partners	6.881% (0.105%)	3.134% (0.118%)	1.743% (0.11%)	1.165% (0.112%)

groups of randomly selected crawlers, and (2) groups of crawlers that are partners of each other.[17] In both cases, we randomly selected 1000 groups of crawlers that meet the given criterion. By comparing the overlap in the lists of partners in these two scenarios, we are able to assess whether groups of clients that are partners tend to share more or less partners than randomly selected clients.

Table 6 shows average results, along with corresponding SEM values in parenthesis, for group sizes varying from 2 to 5. The small SEM values, reported as fractions of the corresponding averages, indicate the small variability of the results across different snapshots and transmissions. Let's consider the randomly selected groups first. As expected, as the group size increases, the overlap in their lists of partners decreases. Whereas for pairs of randomly selected crawlers, the overlap is 6.576%, on average, it drops to 0.1%, on average, for groups of five crawlers. In contrast, the overlap among crawlers that are themselves partners of each other is clearly larger, particularly for larger groups. For instance, pairs of crawlers that are partners share 6.881% of their partners, on average, whereas groups of 5 crawlers, all of them partners of each other, have, on average, 1.65% of their partners in common. These results imply that groups of clients that are partners of each other do tend to have more partners in common than randomly selected clients. Nevertheless, we note that, even for groups of partners, the overlap is not very large (at most 6.881%, on average). Such small overlap might impact the effectiveness of decentralized algorithms (e.g., decentralized reputation mechanisms), as we briefly discuss in Section 8.

7.2. Partnership duration

Similarly to session ON and OFF times, we analyze *normalized* partnership durations, representing the duration of each partnership as a fraction of the remaining ON time of the client session during which it was established. We note that, by taking the *remaining ON time*, we capture both the absolute duration of the partnership and the moment when it was established, considering that the session ON time is known. This will be the case if our characterization results are used to generate synthetic workloads: the session level components (ON time, in particular) are generated first, followed by the partnership level components. We also note that we take as references the ON times measured for the real SopCast clients, characterized in the previous section, as opposed to the ON times of our PlanetLab crawlers which, based on our experimental setup, are equal to the total transmission duration. Thus, in this section, we analyze only partnerships established between a crawler and a real client, disregarding partnerships between two crawlers.

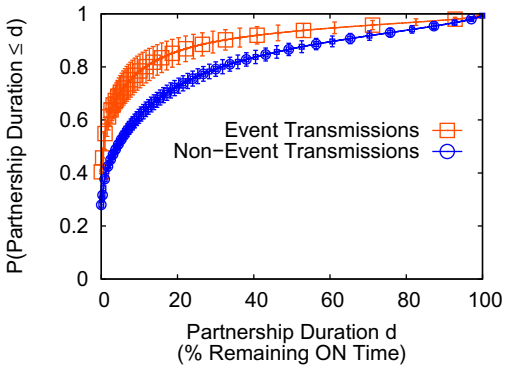


Fig. 10. Normalized partnership durations.

Once again, we found that the distributions of partnership durations are very similar across all transmissions of the same type. Indeed, SEM values computed over the distributions for individual non-event and event transmissions are at most 2.6% and 4.8% of the reported averages, respectively. Although somewhat larger than the SEM values computed for other model components, these values are still reasonably small, implying a small variability in the distributions across different transmissions of the same type.

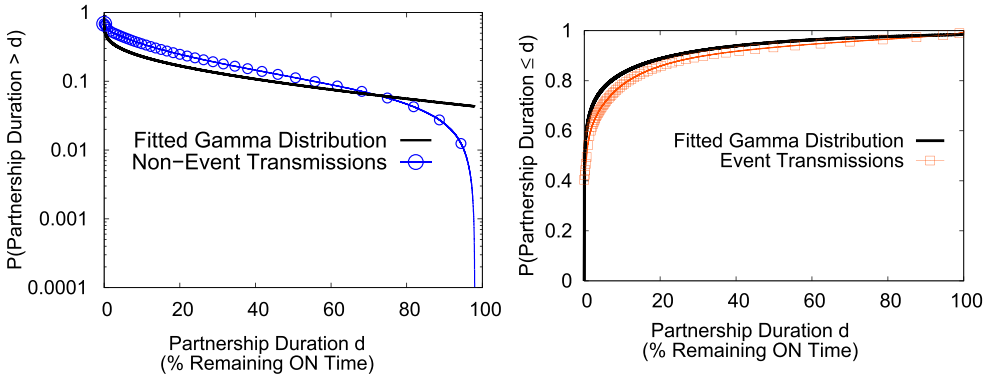
However, we did observe clearly distinct behaviors depending on the transmission type. As shown in Fig. 10, partnerships during event transmissions tend to be much more dynamic, lasting for much shorter periods of time. In particular, we observed that 80% of all partnerships established during event transmissions have durations below 12% of the session remaining ON time. In contrast, during non-event transmissions, only 64% of the partnerships have normalized durations under the same mark.

We can interpret such differences in light of our results for number of partners per client and session ON times. Recall that the distributions of number of partnerships per client, measured at each snapshot, are approximately the same for all event and non-event transmissions. Thus, the normalized duration of each partnership is directly influenced by the duration of the session during which it was established. Since ON times tend to be much longer for event transmissions (see Section 6.4), (normalized) partnership durations tend to be much shorter for that transmission type.

We found that different Gamma distributions are the best fits for partnership durations for event and non-event transmissions, as shown in Fig. 11. To make visual inspection clearer, we plot the curves for non-event transmissions as complementary cumulative distributions (Fig. 11(a)). Table 7 summarizes our findings regarding partnership durations.

Notice that channel popularity might also impact partnership duration. One could expect that in less popular channels clients would have fewer partnership options, and thus would tend to keep their established partnerships for longer periods. However, we note that channel popularity varied quite significantly across different non-event transmissions. Nevertheless, in spite of such differences, the distributions of partnership duration in all non-event transmissions are reasonably similar. These observations provide evidence that channel popularity, although possibly having some influence, might not be a primary factor impacting partnership durations, and client behavior in general. This issue is further discussed in Section 8.

[17] There are partnerships between all pairs of crawlers in the group.



a) Non-Event Transmissions. b) Event Transmissions.

Fig. 11. Normalized partnership duration: best-fitted distributions.

Table 7
Distributions of normalized partnership durations: summary.

Transmission type	Best fit	Mean (%)	Stddev (%)	First parameter	Second parameter
Non-event	Gamma	15.368	24.557	$\alpha = 0.123$	$\beta = 0.123$
Event	Gamma	8.272	19.950	$\alpha = 0.118$	$\beta = 0.700$

8. Summary of our findings and their implications

A summary of our characterization results is provided in Table 8. Based on these findings we make the following key observations.

First, we found close agreement across all monitored transmissions of the same type of content (event or non-event) for all components of our client behavior model, implying that the same distribution provides a reasonably good fit for all such transmissions (and, for non-event transmissions, even for multiple channels). Thus, client behavior follows very similar patterns across different transmissions, provided they are of the same type.

Second, the type of transmitted content (event or non-event) does impact client behavior in terms of session inter-arrival times, ON times and partnership durations. During event transmissions, clients tend to exhibit a more stable behavior, arriving within shorter intervals, mostly at the beginning of the transmission, and remaining active in the system for longer periods of time. We argue that such differences are a direct consequence of the different nature of the two types of content. Recall that event transmissions broadcast major timely events (e.g., final championship matches, decisive political debates). Thus, in comparison with regular non-event transmissions, they transmit content that tends to have more "value" to the interested user, or, in other words, that most likely will be "missed" by users who have some interest in it. Thus, they tend to keep the user's attention for longer periods. Notice that this may not be directly related to the instantaneous popularity of the transmission, in terms of the number of simultaneous clients: a transmission that attracts the interest of a large number of clients may not necessarily keep it for very long (see further discussion below).

These two findings could be exploited in the design of protocols that take the expected client behavior, particularly in terms of dynamic patterns, into account to build the overlay network. For instance, there have been a few proposals of hybrid P2P overlay architectures, which combine tree-based and mesh-based structures [36–38]. One could think of a hybrid strategy that adjusts the structure based on the expected level of client stability in the system, given the type of content that will be transmitted.

Table 8
Hierarchical characterization of SopCast client behavior: summary.

Hierarchy level	Model component	Transmission type	
		Non-event	Event
Session level	Session inter-arrival times	LogNormal distribution	LogNormal distribution
	Number of sessions per client	$\leqslant$2 for 80% of clients	
	ON times	Gamma/ LogNormal distribution	Weibull distribution
	OFF times	Same exponential distribution	
Partnership level	Number of partnerships	Same Gamma/normal distribution	
	Partnership durations	Gamma distribution	Gamma distribution

Our third observation is that, in spite of the differences between event and non-event transmissions, SopCast clients, in general, tend to be very impatient. Most clients have very short ON times. Indeed, around 56% and 79% of clients in event and non-event transmissions, respectively, have ON times under 5 min, whereas 78% and 55% of them have sessions that last for at most 1 min. This is illustrated in Fig. 12, which shows, for each client session during a non-event transmission, the time instants (y-axis) when the session started (continuous line) and when it finished (dot). Thus, one can determine the duration of a session by drawing a vertical line connecting the solid line to a dot. Clearly, most dots appear very close to the line, implying in very short ON times.

Such short ON times reflect a high degree of churn, which may have serious consequences to P2P live streaming effectiveness. As discussed in Section 3, when a client joins the system, it first subscribes to the bootstrap server, which keeps information about the new client (e.g., its network address) to distribute to other clients searching for potential partners. However, the client does not unsubscribe before leaving. Thus, the bootstrap server may keep this information for a while after the client is gone. In our experiments, we did observe a very slow reaction from the SopCast bootstrap server to the departure of a client. Thus, if clients join and leave the system very often, the bootstrap server may end up advertising invalid client addresses as potential partners of joining clients, which will experience longer delays and waste resources until they start receiving the media streaming.

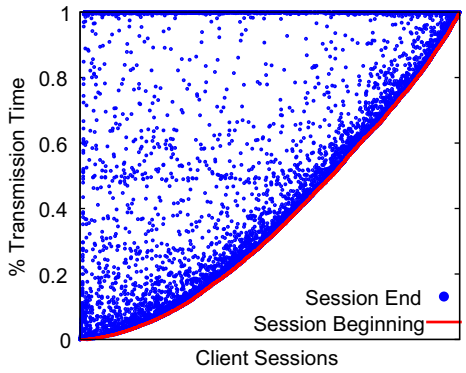


Fig. 12. Client sessions during a non-event transmission.

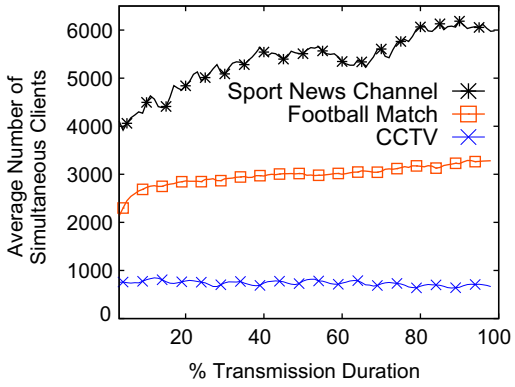


Fig. 13. Popularity across transmissions of different channels.

We also observe that most clients are, in general, very aggressive and inconstant when it comes to establishing partnerships: they drop old partners and add new ones very often. This is evidenced by a large number of short partnerships in most analyzed client sessions during event and non-event transmissions. This result motivates an investigation of the impact of currently used partner selection algorithms on system performance, and possibly the design of strategies that favor more stable and longer partnerships, which might yield better performance. Another relevant finding is the typically small overlap in the list of partners of multiple clients (even clients that are partners), which might impact the effectiveness of decentralized reputation systems (e.g. [8,39]), in which a client estimates the reputation of a future partner based on information received from other clients about their previous interactions with it.

As a final note, we point out that the similar client behavior across transmissions of the same type happens in spite of great differences in channel popularity. To illustrate this point, Fig. 13 shows the number of simultaneous clients connected to each monitored channel as a function of time, during one transmission of each channel. Time is presented as a fraction of the total transmission duration. Clearly, the two channels broadcasting non-event transmissions (i.e., CCTV and Sport News Channel) experience very different popularities throughout the transmission. Whereas the number of clients connected to CCTV remains roughly stable (around 700), the popularity of Sport News tends to increase as the transmission progresses, experiencing peaks of over 6000 simultaneous clients. In comparison, the transmission of a major football match (event) experiences an intermediate popularity, which increases slowly, exceeding 3000 simultaneous clients by the end of the transmission.

We should note that the two non-event transmissions shown in Fig. 13 seem to be two extreme cases for which there might be some differences between the distributions found for each model component. As a matter of fact, as discussed in the previous section, it is intuitive that the number of participants in the system may impact some aspects of client behavior, such as number of partners and partnership duration. However, considering all 35 non-event transmissions, the small SEM values computed over the individual distributions, for all model components, indicate that, in general, such differences tend to be small. Thus, although popularity might impact, to some extent, client behavior, we argue that content type, in terms of event or non-event, is a more determinant factor.

9. Conclusions and future work

This paper presents a characterization of client behavior in SopCast, focusing particularly on dynamic behavior aspects. Our characterization was driven by a hierarchical model that captures client behavior at both session and partnership levels. It was performed separately for various transmissions of major live events as well as regular (non-event) content, which differ particularly in terms of the "value" of the content to the user. For each component of our model, we provided best fitted statistical distributions, which can be used to drive the generation of realistic synthetic workloads. In general, we found that client behavior patterns are quite consistent across all monitored transmissions of the same type of content, whereas session inter-arrival times, session ON times and partnership durations exhibit very distinct patterns for event and non-event transmissions. Thus, the inherently different nature of the two types of content does have a significant impact on client behavior.

Possible directions for future work include characterizing client behavior, according to our hierarchical model, in other P2P live streaming applications, further analyzing the correlation between different model components as well as their impact on P2P protocol design and optimization, and building a realistic synthetic workload generator for live P2P streaming applications. acknoAcknowledgementsThis research is partially funded by the authors' individual grants from CNPq, CAPES e FAPEMIG as well as by the Brazilian National Institute of Science and Technology for Web Research (MCT/CNPq/INCT Web Grant No. 573871/2008–6).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.comcom.2012.02.012.

References

[1] NBC, www.nbc.com (last accessed at June 2011).
[2] ESPN, www.espn.com (last accessed at June 2011).
[3] Z. Xiao, F. Ye, New Insights on Internet Streaming and IPTV, in: Proceedings of International Conference on Content-Based Image and Video Retrieval, 2008, pp. 645–654.
[4] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, S. Tewari, Will IPTV ride the peer-to-peer stream?, in: IEEE Communications Magazine, Vol. 45(6), 2007, pp. 86–92.
[5] Y. Cui, L. Dai, Y. Xue, Optimizing P2P streaming throughput under peer churning, in: Proceedings of IEEE GLOBECOM, 2007, pp. 231–235.
[6] M. Haridasan, R.V. Renesse, SecureStream: an intrusion-tolerant protocol for live-streaming dissemination, in: Journal of Computer Communications, vol. 31(3), 2008, pp. 563–575.
[7] Y. Tang, L. Sun, M. Zhang, S. Yang, Y. Zhong, A novel distributed and practical incentive mechanism for peer to peer live video streaming, in: Proceedings of IEEE International Conference on Multimedia & Expo, 2006.
[8] A. Borges, J. Almeida, S. Campos, Fighting Pollution in P2P Live Streaming Systems, in: Proceedings of IEEE International Conference on Multimedia & Expo, 2008.
[9] I. Chatzidrossos, V. Fodor, On the Effect of Free-Riders in P2P Streaming Systems, in: Proceedings of International Workshop on QoS in Multiservice IP Networks (QoSIP), 2008.

[10] M. Zhang, J.-G. Luo, L. Zhao, S.-Q. Yang, A Peer-to-Peer Network for Live Media Streaming Using a Push-Pull Approach, in: Proceedings of ACM International Conference on Multimedia, 2005, pp. 287–290.
[11] N. Magharei, R. Rejaie, Y. Guo, Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches, in: Proceedings of IEEE INFOCOM, 2007, pp. 1424–1432.
[12] X. Hei, Y. Liu, K.W. Ross, IPTV Over P2P Streaming Networks: the Mesh-Pull Approach, in: IEEE Communications Magazine, Vol. 46(2), 2008, pp. 86–92.
[13] T. Silverston, O. Fourmaux, A. Botta, A. Dainotti, A. Pescapé, G. Ventre, K. Salamatian, Traffic Analysis of Peer-to-Peer IPTV Communities, in: Computer Networks, Vol. 53(4), 2009, pp. 470–484.
[14] T. Silverston, O. Fourmaux, Measuring P2P IPTV Systems, in: Proceedings of NOSSDAV, 2007.
[15] T. Silverston, O. Fourmaux, P2P IPTV Measurement: A Case Study of TVAnts, in: Proceedings of ACM CoNEXT Conference, 2006, pp. 1–2.
[16] L.H. Vu, I. Gupta, J. Liang, K. Nahrstedt, Measurement and Modeling of a Large-scale Overlay for Multimedia Streaming, in: Proceedings of Int'l Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, 2007, pp. 1–7.
[17] X. Hei, C. Liang, J. Liang, Y. Liu, K. Ross, A Measurement Study of a Large-Scale P2P IPTV System, in: IEEE Transactions on Multimedia, Vol. 9(8), 2007, pp. 1672–1687.
[18] D. Stutzbach, R. Rejaie, S. Sen, Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems, in: IEEE/ACM Transactions on Networking (TON), Vol. 16(2), 2008, pp. 267–280.
[19] M. Rocha, M. Maia, I. Cunha, J. Almeida, S. Campos, Scalable Media Streaming to Interactive Users, in: Proceedings of ACM Multimedia, 2005, pp. 966–975.
[20] B. Fallica, Y. Lu, F. Kuipers, R. Kooij, P.V. Mieghem, On the Quality of Experience of SopCast, in: Proceedings of Int'l Conference on Next Generation Mobile Applications, Services and Technologies, 2008, pp. 501–506.
[21] C. Wu, B. Li, S. Zhao, Exploring Large-Scale Peer-to-Peer Live Streaming Topologies, in: ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), Vol. 4(3), 2008, pp. 1–23.
[22] X. Hei, C. Liang, J. Liang, Y. Liu, K.W. Ross, Insights into PPLive: A Measurement Study of a Large-Scale P2P IPTV System, in: Proceedings of IPTV Workshop, in conjunction with International World Wide Web Conference, 2006.
[23] Y. Zhou, D.-M. Chiu, J.C.S. Lui, A Simple Model for Analyzing P2P Streaming Protocols, in: Proceedings of IEEE International Conference on Network Protocols, 2007, pp. 226–235.
[24] D. Stutzbach, R. Rejaie, Understanding Churn in Peer-to-Peer Networks, in: Proceedings of 6th ACM Internet Measurement Conference, 2006, pp. 189–202.
[25] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, M. Bowman, PlanetLab: An Overlay Testbed for Broad-Coverage Services, in: ACM SIGCOMM Computer Communication Review, Vol. 33(3), 2003.
[26] T. Silverston, O. Fourmaux, K. Salamatian, K. Cho, On Fairness and Locality in P2P-TV through Large-Scale Measurement Experiment, in: Proceedings of IEEE GLOBECOM, 2010.
[27] X. Zhang, J. Liu, B. Li, Y.S.P. Yum, CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming, in: Proceedings of IEEE INFOCOM, 2005, pp. 2102–2111.
[28] C. Costa, I. Cunha, A. Borges, C. Ramos, M.M. Rocha, J.M. Almeida, B. Ribeiro-Neto, Analyzing Client Interactivity in Streaming Media, in: Proceedings of 13th International World Wide Web Conference, 2004, pp. 534–543.
[29] E. Veloso, V. Almeida, W. Meira, A. Bestavros, S. Jin, A Hierarchical Characterization of a Live Streaming Media Workload, in: IEEE/ACM Transactions on Networking, Vol. 14(1), 2006, pp. 133–146.
[30] A. Pathak, H. Pucha, Y. Zhang, Y.C. Hu, Z.M. Mao, A Measurement Study of Internet Delay Asymmetry, in: Proceedings of International Conference on Passive and Active Network Measurement, 2008, pp. 182–191.
[31] The Network Time Protocol, www.ntp.org/ (last accessed at June 2011).
[32] L. Vu, I. Gupta, J. Liang, K. Nahrstedt, Mapping the PPLive Network: Studying the Impacts of Media Streaming on P2P Overlays, in: UIUC Technical Report (UIUCDCS-R-2006-275), 2006.
[33] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, John Wiley and Sons, INC, 1991.
[34] S. Tang, Y. Lu, J.M. Hernández, F. Kuipers, P. Mieghem, Topology Dynamics in a P2PTV Network, in: Proceedings of IFIP Networking Conference, 2009.
[35] J. Padhye, J. Kurose, An Empirical Study of Client Interactions with a Continuous-Media Courseware Server, in: Proceedings of NOSSDAV, 1998.
[36] F. Wang, Y. Xiong, J. Liu, mTreebone: A hybrid tree/mesh overlay for application-layer live video multicast, in: Proceedings of IEEE International Conference on Distributed Computing Systems, 2007.
[37] Q. Huang, H. Jin, X. Liao, P2p live streaming with tree-mesh based hybrid overlay, in: Proceedings of IEEE International Conference on Parallel Processing Workshops, 2007.
[38] S. Awiphan, Z. Su, J. Katto, Tomo: a two-layer mesh/tree structure for live streaming in p2p overlay network, in: Proceedings of IEEE Consumer Communications and Networking Conference., 2010.
[39] J. Seibert, X. Sun, C. Nita-Rotaru, S. Rao, Towards securing data delivery in peer-to-peer streaming, in: Proceedings of IEEE International Conference on Communication Systems and Networks, 2010.