

# On Feasibility of P2P On-Demand Streaming via Empirical VoD User Behavior Analysis

Bin Chang, Liang Dai, Yi Cui, Yuan Xue  
Department of Electrical Engineering and Computer Science  
Vanderbilt University  
{bin.chang, liang.dai, yi.cui, yuan.xue}@vanderbilt.edu

**Abstract**—In its current art, peer-to-peer streaming solution has been mainly employed in the domain of live event broadcasting. In such a paradigm, users are required to simultaneously participate the streaming, which yields tremendous bandwidth pool to alleviate the server load. However, little effort has been paid to study the performance gain when peer-to-peer solution is deployed into the domain of Video-on-Demand (VoD) applications, when users have the freedom to access a large pool of media files at their preferred times. Users of VoD applications exhibit file-dependent and time-varying access patterns, which are hard to simulate without realistic guidance from the operational system observation. In this paper, we present an empirical study on the traces collected by the Vanderbilt University media streaming service over a period of 8 months. We pay special attention to peer aggregation around one media file, in which peer-to-peer streaming is able to play an essential role. With this regard, we investigate three key factors: file popularity, request inter-arrival time, and user online duration. Our analysis proves the existence of skewed file popularity, concentrated user requests, and long-enough online duration. Furthermore, through replaying the trace via simulation, we show that peer-to-peer streaming could reduce the server load by as high as 90% over popular files.

## I. INTRODUCTION

Recent years have witnessed the rapid transformation of Internet content from text and images to streaming audio/video signals. A growing number of multimedia-based services, such as online radio and TV, distance education, video-on-demand, video sharing, have emerged and flourished, attracting millions of regular Internet users. Accompanying with the success of media-based web service is the significant cost in server storage space and bandwidth subscription to handle the increasing number of hosted media files and visiting users. YouTube, the most popular video-sharing service, is reported to spend millions of dollars monthly on bandwidth cost.

To address this challenge, which is intrinsic to the conventional client-server model, peer-to-peer streaming technology has been developed and successfully deployed. By utilizing uplink bandwidth of the client (peer) machine, peers are enabled to relay the content to each other, which saves the output bandwidth of the server to its minimum. Today, the peer-to-peer streaming systems mainly serve the applications in the nature of live broadcasting. In these applications, the media content is generated on-the-fly and simultaneously pushed to all receivers. Peers must join the streaming during broadcasting, or they will miss to view the content. Despite the inconvenience, this restriction on user participation suc-

cessfully aggregate all interested peers at the same time, which forms the maximum uplink bandwidth pool.

However, the applicability of peer-to-peer technology in the domain of on-demand streaming remains largely in debate. The challenges root at the much broadened selection space on the user side. First, users are entitled to choose among millions of media files to view, as evidenced in current video-sharing web services. Second, along each chosen media files, interested users are free to view it at any time they prefer. While offering users the appreciated freedom, these two features can greatly dilute the aforementioned bandwidth pool by peers, and thus challenge the feasibility of peer-to-peer streaming.

In this paper, we aim to study the feasibility of peer-to-peer streaming by empirically analyzing the user behavior from the traces collected at the operational VoD streaming system at Vanderbilt University[1]. These traces allow us to have a fine-grained examination over file properties and user access patterns. In particular, we search for the presence of “peer aggregation”, where a number of peers are accessing the same media file simultaneously. This feature is desired by peer-to-peer streaming, in which peers are able to relay content to each other.

However, the emergence of peer aggregation relies on the coexistence of following key factors. First, the distribution of media file popularity must be skewed to ensure high request rate on the popular ones. Second, the request inter-arrival time towards certain media file must be short to create the required burstiness. Third, the duration that a peer stays online must be long enough to overlap with the presence of other peers, hence promoting the peer aggregation.

Our study proceeds following the above sequence. We first study the media file popularity and its variation along time. Then we investigate the distribution of overall user request inter-arrival time, as well as the inter-arrival time on individual media files. After this step, we analyze the duration a user remains online. Our analysis proves the existence of skewed file popularity, concentrated user requests, and long-enough online duration. Finally, through replaying the trace via simulation, we show that peer-to-peer streaming could reduce the server load by as high as 90% over popular files.

The rest of this paper is organized as follows. Sec. II offers background by introducing our trace collection procedure and the basic concept of on-demand peer-to-peer streaming. Sec. III studies the file popularity. Sec. IV studies the user

access pattern in terms of inter-arrival time and online duration. Sec. V presents our simulation study. Sec. VI reviews the related work. We finally conclude at Sec. VII.

## II. OVERVIEW

### A. Trace Set

The traces studied in this paper are collected from the log records of two media streaming servers run by Vanderbilt media service[1]. The traces extend over a 8-month period since May 26, 2006. Each log entry consists of (1) user identification in terms of IP address, (2) requested file by its URL, and (3) time stamps recording the start and stop requests by the user.

In Fig. 1, we show the server load in terms of number of user access. In particular, Fig. 1 (a) shows the request per hour in a 3-day period, and Fig. 1 (b) shows requests per day in the entire 8-month period.

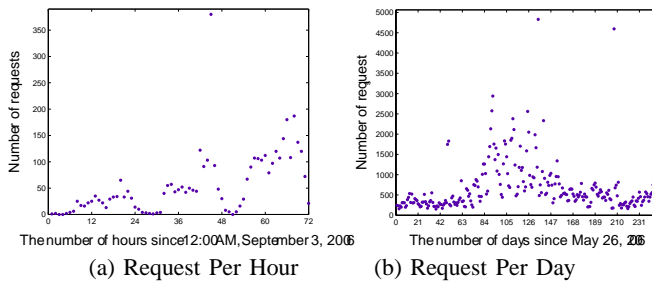


Fig. 1. Server Load

The service supports two of the most popular streaming media formats: Real and Windows Media. A total of 2219 media files have been accessed by 37577 different users as identified by their IP addresses, generating a total of 166793 requests. About 37% of the users only access the streaming service for once.

The media files served by the streaming service fall into three categories: instructional videos including guest lectures, documentaries, and sports events. The bitrates range from 44 Kbps to 1,190 Kbps. The typical bitrates include 44 Kbps, 225 Kbps, and 828 Kbps. Those streaming files with bitrates less than 50 Kbps are audio files.

### B. On-demand Peer-to-Peer Streaming

In Fig. 2, we illustrate the basic concept of how peer-to-peer solution would work in the on-demand video streaming system. A play-and-cache model is employed at each peer, where the content, after being played, is cached at the peer host and relayed to other peers upon request. This is in sharp contrast to the traditional play-and-forward model in live peer-to-peer streaming, where the incoming content is viewed at a peer, forwarded to its children peers and immediately discarded.

This on-demand peer-to-peer streaming model poses additional resource demand on peer's host, i.e., cache space in addition to outbound bandwidth. Besides, it also blurs the time boundary within which a peer is responsible to serve other peers. In live streaming, this period is constrained by

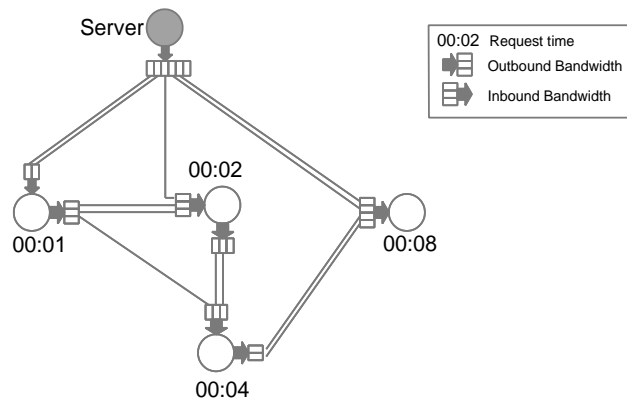


Fig. 2. Illustration of On-demand Peer-to-peer Streaming

the online duration a peer is active inside the system, whereas in on-demand streaming, a peer could extend its serving time for as long as its machine is up and connected.

### C. Methodology

In light of these two new requirements, we use the following methodology in our trace study. First, we define a peer's serving duration to be its online time. Aware that a user could freely view many video files during his/her online time, we pay special attention to the "gap period", which is the time elapsed between the viewing of two video files. If this gap period is within certain threshold, say 60 seconds, we consider the accesses to these two files to belong to the same *online session*. Second, a peer caches only the files it has viewed. Additionally, the cache space is emptied after the user's online session is over, for the purpose of acquitting the peer its relaying responsibility when it becomes offline.

To this end, the essential question we try to study in this work, is whether and to what degree the phenomenon of "peer aggregation" will emerge from the limited cache space maintained by online peers, thus reducing the server load. This phenomenon would not happen if any of the following three key factors are missing: (1) skewed file popularity to ensure high access rate on the popular ones, (2) concentrated request inter-arrival time to grow critical mass of online peers, and (3) long-enough online duration to sustain this peer mass. In what follows, we present our study on these factors by analyzing the traces.

## III. FILE POPULARITY

We sort the popularity of all media files in terms of the number of hits each of them has experienced in the 8-month period. Among the 2,219 files, the top 1% of them have been requested for more than 36,802 times. These popular files together account for 22% of the total number of requests (166,793).

Focusing into each file, we find out that its popularity is also highly bursty along the time. In Fig. 3, we choose representative files and show the number of requests they have experienced per day throughout the 8-month period. File A (play2football2005high.wmv) is sized 11.1MB with

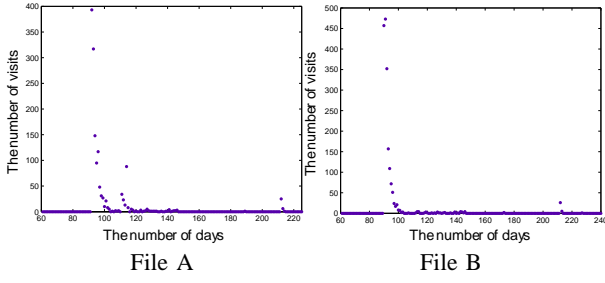


Fig. 3. File Popularity Variation Along Time

bitrate of 225 Kbps, and was requested for a total of 1468 times. As shown in Fig. 3 (a), at its most popular day, file A had almost 400 hits. Within a week time, its popularity gradually dropped to below 50 hits per day. Comparatively, in other days, the request rate is below 10 except two surges around the 115th day and 215th day. Collectively, only 6 days see the number of hits exceeding 50, but they account for 78.8% of all requests along the 8-month time. File B (play2football2005high.wmv) is sized 11.4MB with bitrate of 225Kbps, and was requested for a total of 1827 times. As shown in Fig. 3 (b), it demonstrates a stronger pattern. At its most popular week, the file has experienced at its peak around 480 requests per day, and has concentrated 91.36% of all requests. The remaining days only see requests below 10, except a single surge around the 220th day.

We note that such pattern is only observable in the most popular files. The rest of the unpopular files in the heavy tail can hardly replicate it because of its low overall request rate. Therefore, peer-to-peer solution is most likely to be effective over the most popular files, which evidently account for majority of the server cost in on-demands streaming. In what follows, we focus our study on user access behavior on the two popular files showcased in Fig. 3.

#### IV. USER ACCESS PATTERN

In this section, we study user’s access pattern from the aspects of request inter-arrival time and online session duration. For a group of peers accessing the same file along the timeline, if their online durations are long enough to exceed the inter-arrival time, a “relaying chain” could be formed among them. Further prolonging the online duration would cause the online duration overlapping of more peers, hence the “peer aggregation” desired in peer-to-peer streaming.

##### A. Request Inter-Arrival Time

To study the distribution of request inter-arrival time, we sort all viewing requests in the trace on a chronological manner, then record the inter-arrival time as the time elapsed between consecutive requests. In Fig. 4 (a), we further sort these inter-arrival time instances by their lengths, which exhibits a clear stair-case pattern. Although the maximum inter-arrival time is 35000 seconds, indicating the service is idle for more than 9 hours, 99% of all inter-arrival time instances are within the 1000 seconds interval, which is less than 17 minutes.

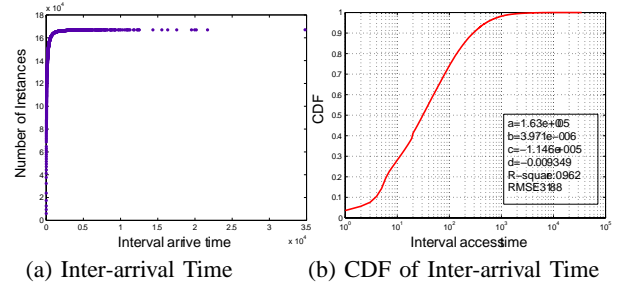


Fig. 4. Inter-arrival Time

In Fig. 4 (b), we plot the cumulative distribution function (CDF) of the above data in log-scale, which reveals more details. 75% of inter-arrival time instances are within the 100 seconds interval, and 30% of instances are within the 10 seconds interval. Also, an almost linear curve extends from 3 seconds to 1000 seconds, suggesting an exponential distribution of inter-arrival times.

Practicing curve fitting in MATLAB confirms this conjecture. We further find the following exponential function to have the best fit. We list the values of the parameters obtained in curve fitting (with 95% confidence bound) in Fig. 4 (b).

$$f(x) = a \cdot e^{b \cdot x} + c \cdot e^{d \cdot x} \quad (1)$$

Next, we analyze the same metric for each media file, where the inter-arrival time is the time elapsed between requests accessing the same file. In Fig. 5, we choose file A and B showcased in Fig. 3, and show the CDFs of their inter-arrival times in log scale.

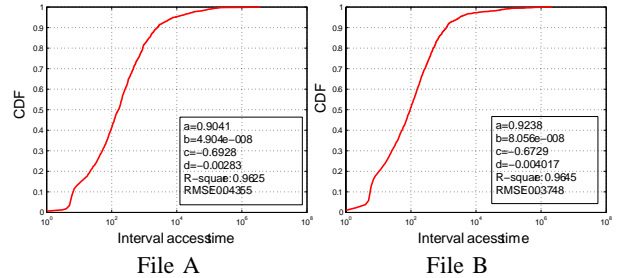


Fig. 5. CDFs of Inter-arrival Times for Selected Files

Although the requests for each file is impossible to be as dense as the requests for all files (Fig. 4), Fig. 5 shows both of them have 85% of the inter-arrival time instances within the 1000 seconds interval. Also, in both files, we find the close to linear relationship from the 10% to 90% percentiles. Therefore, the exponential function in Eq. (1) still gives the best fit. We include the results of our curve fitting in Fig. 5.

Our observation on the distribution of inter-arrival times (both overall and per-file) coincides with the time-varying file popularity observed in Fig. 3. It reveals that user access behavior is bursty both in general and per-file. This further gives hope for peer-to-peer streaming to take effect during the period of high user access rate, i.e., low inter-arrival time.

## B. Online Duration

In Fig. 6 (a), we sort the online durations of all sessions. We show its CDF in Fig. 6 (b). We experiment with different gap period threshold values  $\tau$ .  $\tau$  defines the upper bound time limit, within which the same user's consecutive requests are merged into one session.  $\tau = 0$  denotes the most conservative measure, which regards each user request as an individual session. We also set  $\tau = 100$  and  $\tau = 600$ , which correspondingly allow the same user's requests to be apart from each other for at most 100 seconds and 10 minutes, but still considered to belong to one session. We have the following observations.

First, the distribution of online duration is revealed to be exponential. In the log-scale CDF figure shown in Fig. 4 (b), a linear relationship exists from the beginning to 80% percentile for the case of  $\tau = 0$ , also from 20% to 90% percentile when  $\tau = 100$  and 600. MATLAB curve fitting shows Eq. (1) to be the best fit again.

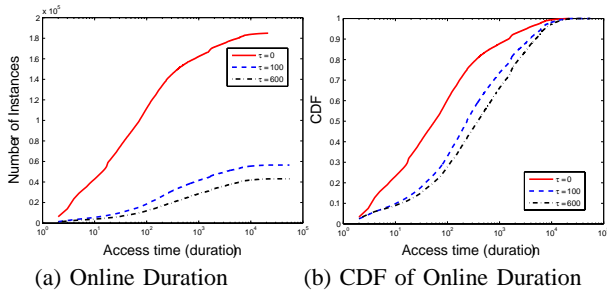


Fig. 6. Online Duration

Second, increasing  $\tau$  proves to greatly prolong the online duration. While this is not as obvious in Fig. 6 (a), mainly due to the fact that merging reduces the total number of requests, the effect is much clear in the CDF figure (Fig. 6 (b)). For example, at 50% percentile, the online duration is 237 seconds for  $\tau = 100$ , and 381 seconds for  $\tau = 600$ , an almost order-of-magnitude improvement compared to the case when  $\tau = 0$  (50 seconds).

Finally, by comparing the CDFs of inter-arrival time (Fig. 4 (b) and Fig. 5) and online duration (Fig. 6 (b)), we discover that the online duration exceeds the inter-arrival time at all percentiles. We consider this final finding the most significant one in this work, since it statistically affirms the existence of “relaying chain” among peers.

We summarize our findings in Tab. I, which compares the time lengths of inter-arrival time and online durations with different  $\tau$  from percentile 10% to 90%. At each percentile level, the online duration is shown to overshadow the inter-arrival time, even when  $\tau = 0$ . If  $\tau$  is extended, the effect becomes more significant, e.g., online duration of 3738 seconds ( $\tau = 100$ ) versus inter-arrival time of 300 seconds at the 90% percentile.

For individual files A and B, due to the diluted inter-arrival time, which the conservative online duration ( $\tau = 0$ ) is not able to cover. However, extending  $\tau$  can result in longer duration able to well cover the inter-arrival times for both files, e.g., online duration of 2329 seconds ( $\tau = 600$ ) versus inter-arrival time of 485 seconds for file B at the 80% percentile.

Such a high ratio might statistically imply considerable peer aggregation.

## V. SIMULATION STUDY

However, the alignment of inter-arrival time and online duration along the CDF percentile does not in fact guarantee their occurrences in the same request instances. To accurately quantify the exact extent of peer aggregation and its impact on server load, we develop a discrete-event simulator. By replaying the arrival processes and their online durations as recorded by the traces, the simulator is able to compute, for each file, the reduced server load via peer-to-peer relaying.

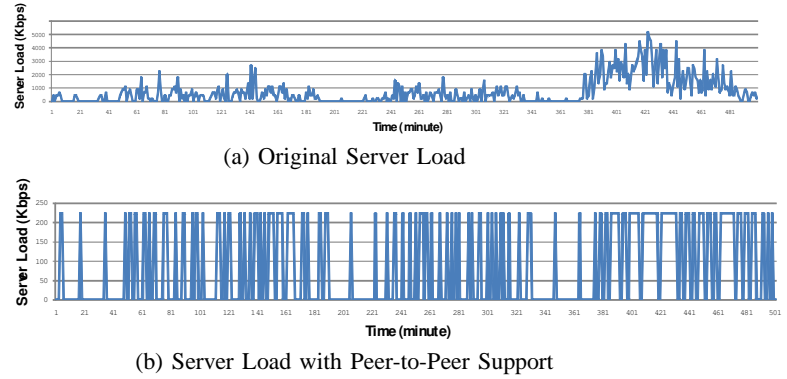


Fig. 7. Comparison of Server Loads Serving marchinginvitational.rm ( $\tau = 0$ )

In Fig. 7, we show the original server load trace when serving the file `marchinginvitational.rm` during its most popular 8 hours, and compare it with the simulated server load if peer-to-peer relaying is in place. Here, we assume that each peer has enough outbound bandwidth to serve at least one other peer. Also we set  $\tau = 0$ , i.e., which treats each request as an individual session.

As shown in Fig. 7 (a), at its peak, the file was requested by up to 20 simultaneous requests per minute, which provides opportunity to form a relaying chain among these requesting peers. Consequently, in Fig. 7 (b), the server load is either at 225Kbps serving only the beginning peer of the chain, or 0Kbps when no requests are present. During the entire 8-month period, the server has sent out 23.93389 gigabits to serve the file. In the setting of our peer-to-peer simulation, the server would send out 2.441812 gigabits, achieving nearly 90% of server load reduction.

## VI. RELATED WORK

A number of existing literatures [2], [3], [4], [5], [6] have been dedicated to the analysis of multimedia workload in terms of file popularity, user arrival process, inter-arrival time, etc. Acharya et al [2] studied the six-month trace data from a multicast VoD system and observed high temporal locality of user accesses and a non-Zipfian distribution of video title ranking. Almeida et al [3] analyzed education media server workloads. The work by Chesire et al [5] focused on session duration, file popularity, sharing patterns among clients, and discovered the popularity distribution to follow Zipfian distribution. Yu et al [6] studied the trace of a national-scale VoD system,

Percentile	Inter-arrival Time (s) (All Files)	Inter-Arrival Time (s) (File A)	Inter-Arrival Time (s) (File B)	Duration (s) ( $\tau = 0$ )	Duration (s) ( $\tau = 100$ )	Duration (s) ( $\tau = 600$ )
10	4	7	6	4	10	13
20	6	22	11	8	38	53
30	10	52	26	17	85	115
40	20	93	53	30	146	212
50	30	161	95	57	237	381
60	40	256	157	98	401	694
70	80	464	268	171	777	1298
80	160	897	485	356	1637	2329
90	300	2550	1230	1486	3738	4939

TABLE I  
COMPARISON OF INTER-ARRIVAL TIME AND ONLINE DURATIONS AT DIFFERENT PERCENTILE

and proposed a modified Poisson distribution model on user arrival process. Finally, several workload generators have been proposed, namely GISMO[7] and MediSyn[8], to synthesize trace by defining individual session characteristics and request arrival process as observed in their own trace studies[9], [10]. Inspired by these previous works, we investigate user behaviors around a single media file in search for the occurrence of “peer aggregation”. We discovered that the distributions of inter-arrival time and online duration regarding the most popular individual files resemble great similarity to the same distributions observed over all files.

In the peer-to-peer domain, many measurement works have been conducted to study overlay topology in Gnutella file sharing[11], file popularity and peer locality in Kazaa system[12], peer population and heterogeneity in Napster[13], etc. Perhaps the one with the most similarity with our work with regard to research objective and methodology, is by Huang et al[14]. In this work, traces from MSN VoD service is collected to study the server load reduction via the assistance of peer-to-peer streaming.

## VII. CONCLUSION AND FUTURE WORK

In this work, we research on the feasibility of on-demand peer-to-peer streaming through empirical evaluation on traces collected from the Vanderbilt streaming media service. In particular, we search for “peer aggregation” by looking at three key factors: file popularity, inter-arrival time, and online duration, which collectively nurture the emergence of this phenomenon. Our analysis proves the existence of skewed file popularity, concentrated user requests, and long-enough online duration. Furthermore, through replaying the trace via simulation, we show that peer-to-peer streaming could reduce the server load by as high as 90% over popular files.

Continuing on these encouraging results, we will further refine our study along several directions. For example, our simulation assumes the relaying chain formed among peers to be linear. In reality though, given the heterogeneity of peer outbound capacities, some might be powerful enough to serve multiple peers, some might not be able to serve a single peer. By collecting IP addresses of all peers, we could estimate their outbound bandwidths by leveraging broadband speed testing service such as broadbandreports.org, thus more accurately quantify the amount of bandwidth collected through peer aggregation. Other directions might include taking firewall

into consideration and studying the impact of peer-to-peer streaming on inter-ISP traffic.

## REFERENCES

- [1] “Vanderbilt streaming service,” <http://its.vanderbilt.edu/video/archives.php>
- [2] S. Acharya, B. Smith, and P. Parnes, “Characterizing user access to videos on the world wide web,” in *Multimedia Computing and Networking (MMCN)*, January 2000.
- [3] Jussara M. Almeida, Jeffrey Krueger, Derek L. Eager, and Mary K. Vernon, “Analysis of educational media server workloads,” in *NOSSDAV*, 2001.
- [4] J. Padhye and J. Kurose, “An empirical study of client interactions with a continuous-media courseware server,” in *NOSSDAV*, 1998.
- [5] Maureen Chesire, Alec Wolman, Geoffrey M. Voelker, and Henry M. Levy, “Measurement and analysis of a streaming media workload,” in *USITS*, 2001.
- [6] Hongliang Yu, Dongdong Zheng, Ben Y. Zhao, and Weimin Zheng, “Understanding user behavior in large-scale video-on-demand systems,” in *EuroSys*, 2006.
- [7] Shudong Jin and Azer Bestavros, “Gismo: A generator of internet streaming media objects and workloads,” in *ACM SIGMETRICS Performance Evaluation Review*, November 2001.
- [8] Wenting Tang, Yun Fu, Ludmila Cherkasova, and Amin Vahdat, “Medisyn: a synthetic streaming media service workload generator,” in *NOSSDAV*, 2003.
- [9] Eveline Veloso, Virgilio Almeida, Jr. Wagner Meira, Azer Bestavros, and Shudong Jin, “A hierarchical characterization of a live streaming media workload,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 133–146, 2006.
- [10] Ludmila Cherkasova and Minaxi Gupta, “Characterizing locality, evolution, and life span of accesses in enterprise media server workloads,” in *NOSSDAV*, 2002.
- [11] Daniel Stutzbach, Reza Rejaie, and Subhabrata Sen, “Characterizing unstructured overlay topologies in modern p2p file-sharing systems,” *IEEE/ACM Transactions on Networking*, 2007.
- [12] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan, “Measurement, modeling, and analysis of a peer-to-peer file-sharing workload,” in *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, New York, NY, USA, 2003, pp. 314–329, ACM Press.
- [13] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble, “A measurement study of peer-to-peer file sharing systems,” in *Multimedia Computing and Networking (MMCN)*, January 2002.
- [14] Cheng Huang, Jin Li, and Keith Ross, “Can internet video-on-demand be profitable,” in *ACM SIGCOMM*, 2007.