# Understanding the Roles of Servers in Large-scale Peer-Assisted Online Storage Systems

Fangming Liu[†], Ye Sun[†], Bo Li[*†], Xinyan Zhang[§]
[†]*Hong Kong University of Science & Technology*, [§]*Roxbeam Inc.*

***Abstract*—Online storage systems that provide versatile and convenient platforms for content distribution have attracted significant attention over the Internet. To guarantee adequate levels of service quality and to minimize server cost, such systems typically deploy dedicated servers while effectively utilizing peer bandwidth in a complementary fashion. It is essential to understand the role of servers and critical factors that influence the server contributions.**

**In this paper, with full knowledge of internal mechanisms of a large-scale peer-assisted online storage system, namely *FS2You*, and large set of real-world traces, we examine the role of servers in such a system. Specifically, through analyzing server traffic volumes versus various critical factors including file popularity, time period (both "cold" and "hot" periods), and peer types, we not only reveal empirical observations that are contrary to general belief with in-depth rationales, but also exploit potential flaws of current design and strategy, which further draw practical implications on future design.**

## I. INTRODUCTION

Online storage systems, also referred to as one-click hosting, generally provide web services that allow Internet users to easily upload files onto dedicated servers. Most such services simply return a URL that can be shared to others, who can then download the file at a later time. Due to the simplicity and versatility of its user interface, this type of file sharing has rapidly become a favorite among users, overtaking well-known peer-to-peer (P2P) file sharing services, such as BitTorrent.

As online storage systems become increasingly popular, however, server capacity costs have become prohibitively expensive. *Rapidshare*, one of the most well-known one-click hosting systems, deployed a total of 1500 terabytes of online storage in its data centers, in Asia alone. Skyrocketing costs from server-based architectures have motivated the natural idea of taking advantage of peer bandwidth contributions to mitigate server capacity costs. It is nevertheless non-trivial to design and fine-tune a new system that utilizes peer bandwidth in a complementary fashion, without sacrificing the ease of use, reliability, and performance of one-click hosting.

In response, FS2You [1], a peer-assisted online storage system, has been implemented to dramatically mitigate server capacity costs, while maintaining the ease of use and performance comparable to the best server-based solutions. Since the launch of FS2You, it has quickly become one of most popular online storage systems in mainland China. However, little is known about the crucial and practical questions such as to what extent have the server capacity been utilized and saved; what are the performance implications in current design and strategy; what are the potential improvements that can be drawn from empirical experiences.

In this paper, with full knowledge of architecture and protocols of FS2You, and a large set of real-world traces that we have collected, we examine the role of servers and critical factors that influence the server contributions. Specifically, through analyzing server traffic volumes versus various essential features including file popularity, time period (both "cold" and "hot" periods), and peer types (behind NAT or direct-connect), our major contributions are: (1) we demonstrate that the system, while conserving substantial server capacity, is able to scale to a large number of users, and withstand the test of a tremendous volume of traffic over a long period of time; (2) we demonstrate that server contribution, especially the peak server stress, is strongly correlated with file popularity and time evolution, as well as the percentage of peers behind NAT involving in the system; (3) we expose flaws of current design and strategy with concrete evidences and analysis, which draw practical implications that are desired to be judiciously considered to avoid prohibitive peak server stress.

While we recognizing the significance of prior research efforts on measurement studies of existing P2P file sharing systems [2]–[5], there exist important differences between P2P file sharing and peer-assisted online storage systems, particularly regarding the server involvement with complementary peer assistance. With respect to study on the server related issues, Das *et al.* [6] have discussed through an analytical model the motivation and effects of server participation in BitTorrent-like P2P file sharing systems. Wu *et al.* [7] have proposed based on real-world traces an online server capacity provisioning algorithm for multi-channel live P2P streaming systems. Our work is substantially different from these works. To the best of our knowledge, this paper represents a first attempt in the literature to multi-dimensionally investigate the server contributions in a real-world peer-assisted online storage system with full knowledge of the internal mechanisms.

## II. SYSTEM ARCHITECTURE AND COMPONENTS

The FS2You system comprises the following major components: (1) *Directory Server*: each file referred as a *channel* is assigned with a unique channel ID. A directory server keeps the information of all channels (files) including the channel ID and the MD5 (Message-Digest algorithm 5, the hash value of the corresponding file); (2) *Tracking Server*: this maintains the participating peers' information for each channel; (3)

---

[*]The research was support in part by grants from RGC under the contracts 615608, 616207 and 616406, by a grant from NSFC/RGC under the contract N_HKUST603/07, by a grant from HKUST under the contract RPC06/07.EG27.

*Replication Servers*: FS2You deploys 60 replication servers in China. Replication and content distribution mechanisms will be described later in Sec. II-C; (4) *Peers*: groups of peers with protocols to interact with servers, as well as communicate among themselves to share interested content.

### A. Peer Partnership and Overlay Management

All peers involved in a file, *i.e.,* either downloading the file or holding a replica of the file, are organized into an *overlay* for exchanging block availability information and content sharing.

To bound the control overhead of overlay management, FS2You combines coarse-grained tracking servers and decentralized gossip methods for constructing and managing overlays. When a downloading peer joins FS2You, it contacts the directory server and tracking server, and obtains a list of 20 randomly selected peers associated with the same channel. These peers become the initial *partners* of the newly arrived peer, in which this partnership is periodically updated (every 5 minutes).

To maintain accurate lists of peers in each channel on the tracking server, peers report their status to the tracking server every 5 minutes, which contains vital peer information such as a unique peer identifier, its IP address, and information about channels that it has joined. Upon receiving status reports from peers, the tracking server periodically updates the corresponding list of peers associated with each channel. Such a periodic refresh of peer lists in each channel (associated to each file) assists peers to gain access to active partners that are most helpful, with a reasonable level of overhead. Consequently, the download performance can be improved, and the load on servers can be alleviated.

### B. Content Delivery

Each file is divided into fixed size blocks of 256 KB. A *Block Map* (BM) is introduced to specify the availability of blocks at each peer [8]. The periodic exchange of BMs among peers enables them to locate the needed blocks. Each peer can retrieve distinct blocks from its partners simultaneously.

FS2You implements a unique *sequential block scheduling* mechanism, which is timer-driven with a period of 5 seconds; in each period, peers sequentially request missing blocks up to the number of the current partners. In order to minimize the server involvement for cost saving, peers are allowed to request blocks from servers, only under the conditions: (1) peers have insufficient number of partners; (2) none of the partners hold the desired block; (3) the aggregate download rate from partners falls below 10 KB/second. This threshold is empirically determined to prevent peers from aggressively consuming server capacity.

### C. Server Strategies

Servers in FS2You not only provide online storage, but also cooperate with content distribution. Several strategies are adopted to facilitate content replication and distribution.

*Uploading service.* In FS2You, users are allowed to upload a variety of files to servers without *any* size or format limitations. This attracts millions of users to upload a huge volume of content to FS2You, catapulting it to one of the most popular online storage systems in China in a short period of time. Our measurements showed that 500 GB to 1 TB of files are routinely uploaded per day. To cope with such a demand without consuming excessive resources, the following three strategies are adopted: (1) When a user requests to upload a file, the system ensure that only one copy is stored in one of the servers; (2) this copy is stored in the server nearest to the user requesting to upload. This helps to reduce the uploading time, and to mitigate unnecessary cross-AS traffic; (3) with a limited pool of server storage, files of less popularity or/and large sizes are selected for removal from servers.

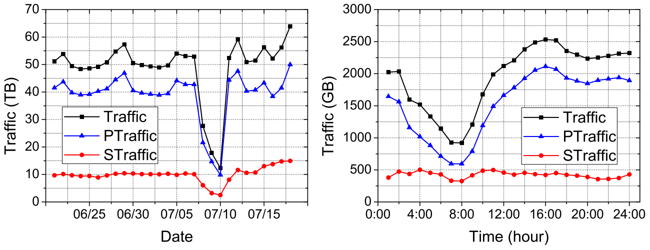
*Downloading service.* Servers complement peers to supply file blocks, especially to those peers suffering poor downloading rates, *e.g.,* below 10 KB/second. The challenge, however, is how to properly satisfy the potentially large number of requests without incurring prohibitively high bandwidth costs. In FS2You, when a server receives a block request, it makes its decisions based on the following policies: (1) The request will be served in a probabilistic fashion, based on the popularity of the file. Specifically, a *file popularity index* is computed for each file periodically, which is inversely proportional to the number of references for this file during the previous period. The rationale behind this is that, a larger number of references will likely result in more replicas of the file at different peers. This simple policy enables peers interested in popular files to largely rely on peer assistance, while dedicating server capacity to less popular files. (2) During a "cold" period with fewer peers, typically from 2 a.m. to 4 a.m., the server will satisfy all the requests for blocks without using the probabilistic serving strategy. In our forthcoming measurement studies in Sec. III, we will examine how the above policies influence the server contributions.

## III. MEASUREMENT RESULTS AND ANALYSIS

In this Section, we take advantage of the large volume of real-world traces that we have collected from over three million users to demonstrate the overall scale and performance of FS2You, and then concentrate on the analysis of server contributions versus various critical factors.

### A. Log and Data Collection

In order to evaluate and analyze the performance of FS2You, we have implemented FS2You with a detailed logging mechanism. Each peer in FS2You is designed to report its activities and status to the trace server, using the HTTP protocol. The trace server appends the time of receipt to each report, and then store it locally in log files. Specifically, a *Download Event Summary* records important statistics between the time when a peer opens a channel (*i.e.*, starts downloading), and when the peer closes the channel (*i.e.*, completes or aborts downloading). The *summary* captures the following: (1) the peer and channel IDs; (2) the size of the file being downloaded; (3) the amount of data downloaded so far; (4) the time instants when the peer opens and closes the channel; (5) the time of



(a) (b)

Fig. 1. Overall scale and performance of FS2You: (a) the total traffic (Traffic), P2P traffic (PTraffic) and total server traffic volumes (STraffic) of FS2You from June 21 to July 18, 2008; (b) the evolution of total traffic (Traffic), P2P traffic (PTraffic) and total server traffic volumes (STraffic) of FS2You over time on July 2, 2008.

the download completion; and (6) the amount of data that are directly served by servers, rather than by peers.

We collected 350 GB traces from 3.3 million users June 21 to July 18, 2008. It is inevitable that there could be inaccuracies in our instrumentation and trace collection mechanism, due to clock skew, crash failures of peers or/and trace server. For example, the trace server indeed crashed and service was interrupted on three days in July. We are convinced that the large volume of traces is valuable even with such imperfections due to real-world complications, as we shall demonstrate in the next subsection.

### B. Overall Scale and Performance

We have totally captured $3,384,948$ online peers over a one-month period from June 21 to July 18, 2008. Fig. 1(a) shows the amount of traffic over the month, where *STraffic* stands for the total traffic volumes served by 60 replication servers (henceforth referred to as *total server traffic volumes*), *PTraffic* represents the traffic contributed by peers, and *Traffic* is the sum of STraffic and PTraffic. The sharp decrease is attributed to the crash failures of the trace server (Sec. III-A). We have made the following observations. (1) The total Traffic stayed around 49 TB to 55 TB during weekdays and reached its peek around 55 TB to 65 TB during weekends. This is related to an observation that there were relatively more active peers with download demand during weekends, thereby leading to higher traffic; (2) Compared to the total Traffic, the total server traffic volumes were fairly stable and stayed around 10 TB; (3) Over the entire month, up to 80% of the traffic was contributed by P2P delivery, which significantly offloaded the server.

A closer look at the daily traffic evolution of FS2You on a representative day is shown in Fig. 1(b). From 8 a.m. to 2 p.m., there is a steady rise of traffic as an increasing number of users join the system, and the *P2P efficiency* (PTraffic / Traffic) increases from 70% to 85%. Specifically, even during a "cold" period (6 a.m. to 8 a.m.) with fewer users, our design of peer assistance can successfully conserve more than 70% of the server capacity. For the remainder of the time, the P2P efficiency steadily stayed around 80% and reached its peak of 85.7% at 10 p.m.

In summary, these measurements have testified that our architectural and protocol designs in FS2You can indeed scale to a large number of peers, and to withstand the test of a tremendous volume of traffic (in the order of terabytes per day) over a long period of time. It is evident that the cost of server capacity has been substantially saved by peer assistance.

### C. Impact From File Popularity

To obtain a fine-grained understanding of server contributions, we now focus on a representative day (July 2, 2008) with over $570,000$ of requests for a variety of files. Fig. 2 plots the cumulative distribution of the file request count and the corresponding total traffic and total server traffic volumes, versus the descending order of file popularity (ranked by request count). We found that: (1) the curve corresponding to total server traffic volumes is less skewed than that of both the requests and total traffic, implying that peer assistance effectively mitigated the server load accounted for by the most popular files; (2) however, popular files still consumed larger portion of server capacity compared to less popular ones. Specifically, the top 25% popular files made up around 62% of server capacity costs.

We further examine the total server traffic volumes by taking into account both the file popularity and time period. Specifically, we classify the observed files into three categories based on the number of requests and replicas: (1) *popular files* are defined as the top 10% of files with larger number of requests within the measured day; (2) among the remaining 90% of files, those with more than 60 replicas (i.e., the average number of replicas of the remaining 90% of files) are regarded as *semi-popular files*; while the remaining are regarded as *unpopular files*. The semi-popular files are often files that were popular in previous measured periods with sufficient replicas among peers.

Fig. 3 plots the total server traffic volumes for three categories of files within a 24-hour period. Fig. 4 plots the *request-to-replica* ratio (defined as the number of requests divided by the number of replicas) of the three categories of files. We have discovered the following:

*First*, during "cold" period such as from 2 a.m. to 8 a.m., the total server traffic volumes accounted for by popular files first increase from 175 GB to a peak of 250 GB at 4 a.m.; then drops back to 170 GB at 8 a.m. Such significant variation of total sever traffic volumes is attributed to the temporary waiver of server-side probabilistic serving strategy from 2 a.m. to 4 a.m. (Sec. II-C). More importantly, the peak server stress of 500 GB happens at 4 a.m. as well. These together not only evidence the important role of server strategy for bounding server capacity cost, but also reveal a natural trade-off between user experience and server capacity. While the temporary waiver of server strategy attempts to improve file availability and downloading performance, it brings vital threat to overwhelm servers even during the seemingly safe period with fewest peers. We will further confirm the danger as we take a closer look at peers behind NAT in the next subsection. On the other hand, the total server traffic volumes accounted for by unpopular and semi-popular files both decrease as there are rare demand for them during "cold" period.

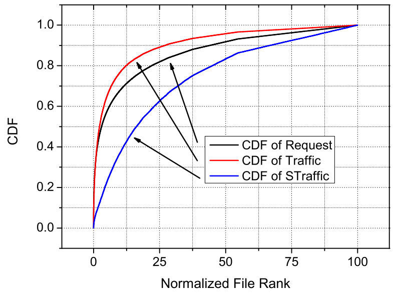


Fig. 2. Cumulative distribution of file request count, the resulting traffic and total server traffic volumes vs. the descending order of file popularity rank on July 2, 2008.

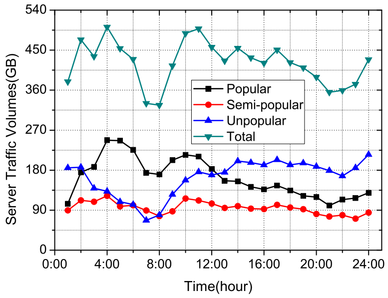


Fig. 3. Total server traffic volumes for three categories of files within a 24-hour period.

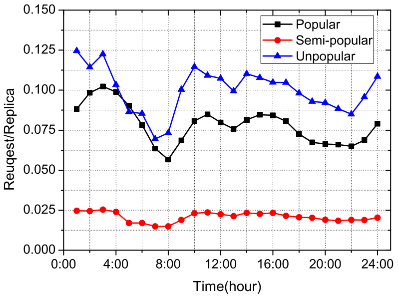


Fig. 4. Request-to-replica ratios of three categories of files within a 24-hour period.

*Second*, during the transition period (such as from 8 a.m. to 10 a.m.) between "cold" and "hot" periods, all the request-to-replica ratios of the three categories of files increase significantly as an increasing number of peers with file requests join the system. This as one aspect results in the rise of total server traffic volumes from all the three categories of files, and consequently causes another peak server stress at 11 a.m. Another aspect relevant to peers behind NAT will be discussed in the next subsection.

*Third*, during "hot" period such as from 10 a.m. to 11 p.m., the total server traffic volumes accounted for by popular files decrease remarkably along the time, which demonstrates the effectiveness of both the server-side probabilistic serving strategy and peer assistance. On one hand, the *file popularity indices* of popular files become larger as time evolves, which results in lower probability to obtain help from servers, and thus implicitly directs peers to rely on peer assistance. On the other hand, as shown in Fig. 4, the request-to-replica ratio of popular files decreases during "hot" period, which implies that the demand for popular files can indeed be satisfied by peer assistance. Conversely, the total server traffic volumes accounted for by unpopular files steadily keep at a high level of around 200 GB during "hot" period, which dominates the majority of entire costs. This reveals that more server resources are allocated to unpopular files with fewer peers; meanwhile, the less demand for a large number of unpopular files renders the server-side probabilistic serving strategy to hardly take effect. With respect to semi-popular files, the resulting total server traffic volumes stay at a relatively lower level of 75 GB to 100 GB. The rationale is that semi-popular files are able to take advantage of sufficient replicas to satisfy relatively smaller number of requests (confirmed by the lower request-to-replica ratio of semi-popular files as depicted in Fig. 4), thus posing less burden on servers.

In summary, server contribution is strongly correlated with file popularity and time period. Though the total server traffic volumes for popular files have been effectively reduced during "hot" period by current design and strategy, there still exists space to reduce the total server capacity cost by cutting down on the portion accounted for by a large number of unpopular files. Further, since the temporary waiver of server strategy during "cold" period brings adverse effects on server capacity cost, a more reliable strategy is desired in future operation.

### D. Impact From Peer Types

We next examine the total server traffic volumes from another perspective by dividing it into portions that consumed by peers behind NAT (henceforth referred to as *NAT peers*) and by direct-connect peers. Among the entire set of observed requests, NAT peers make up a dominant portion of 74%, which reflects the uneven distribution of FS2You user types. Tab. I compares the traffic (from servers, NAT peers, and direct-connect peers) that consumed and contributed by NAT peers and direct-connect peers, respectively. We found that: (1) The small portion of direct-connect peers contributed 50% of traffic, while the large portion of NAT peers only contributed 30% of traffic. This is likely due to the common belief that peers behind NAT are usually hard to be connected by other peers, which restricts the utilization of bandwidth capacities of NAT peers. (2) 21% of the total traffic (39.7 GB) consumed by NAT peers is obtained from servers, which is larger than that of direct-connect peers. This implies that NAT peers are relatively more likely to encounter difficulty in downloading blocks from other peers, thus resort to request help from servers.

TABLE I
TRAFFIC STATISTICS OF NAT PEERS AND DIRECT-CONNECT PEERS.

| | NAT | direct-connect | Total |
|---|---|---|---|
| From Server (TB) | 8.25 | 2.02 | 10.27 |
| Per. | 21% | 16% | 20% |
| From NAT (TB) | 10.28 | 5.72 | 16.00 |
| Per. | 26% | 44% | 30% |
| From direct-connect (TB) | 21.16 | 5.23 | 26.39 |
| Per. | 53% | 40% | 50% |
| Total (TB) | 39.70 | 12.98 | 52.68 |

Based on the above analysis, we believe that the percentage of NAT peers has significant impact on the overall system supply capacity (in terms of bandwidth resources), which in turn can affect the server capacity cost. Intuitively, high percentage of NAT peers with under-utilized bandwidth capacity consume more resources from the system while contribute less to the system. As a result, more server capacity is required by peers to compensate. We will further demonstrate this. Fig. 5 plots the total server traffic volumes for NAT and direct-connect

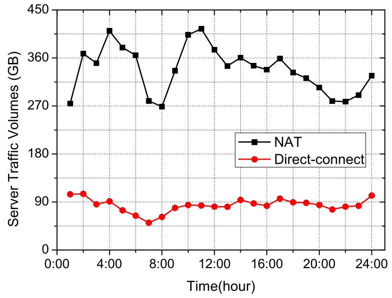


Fig. 5. Total server traffic volumes for NAT peers and direct-connect peers within a 24-hour period.

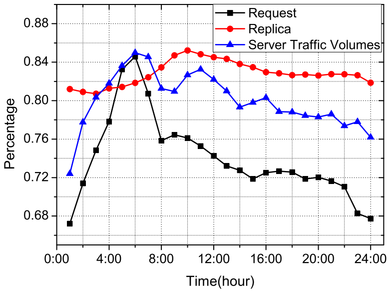


Fig. 6. Percentages of requests, replicas, and total server traffic volumes accounted for by NAT peers, within a 24-hour period.

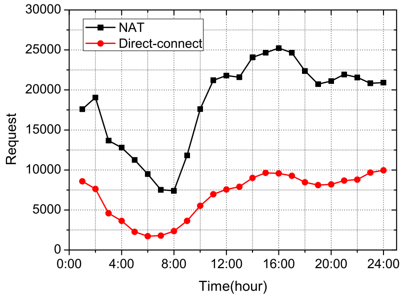


Fig. 7. Number of requests from NAT peers and direct-connect peers within a 24-hour period.

peers within a 24-hour period. Fig. 6 plots the percentages of requests, replicas, and total server traffic volumes for NAT peers, within a 24-hour period. We found the following.

*First*, the total server traffic volumes that consumed by NAT peers make up a dominant portion, and have a larger variation along the time than that of direct-connect peers. Specifically, Fig. 6 shows that more than 78% of total server traffic volumes are consumed by NAT peers over most of the time, and the percentage even increases to around 85% during “cold” period.

*Second*, during “cold” period from 2 a.m. to 6 a.m., though the number of requests from both types of peers drop as shown in Fig. 7, the percentage of requests from NAT peers in Fig. 6 actually increases dramatically! As we discussed previously, the increasing percentage of NAT peers can bring negative effects to the overall system supply capacity. This implicitly drives peers suffering poor downloading performance to request help from servers. Meanwhile, due to the temporary waiver of server-side probabilistic serving strategy from 2 a.m. to 4 a.m., servers are allowed to meet such demand as much as possible. These actually expose the in-depth reason of the peak server stress at 4 a.m., and again confirm the potential danger during “cold” period.

*Third*, during the transition period from 8 a.m. to 10 a.m., Fig. 7 shows that the amount of requests from both types of peers increases; meanwhile, note that the percentage of replicas accounted for by NAT peers increases as well and reaches its peak of 85% at 10 a.m., as depicted in Fig. 6. This implies that while an increasing number of peers with file requests join the system, a majority of replicas are actually hold by NAT peers, and thus under-utilized to supply the demand. This is another aspect that leads to the second peak server stress at 11 a.m.

*Finally*, as shown in Fig. 6, during the “hot” period there are relatively higher percentage of direct-connect peers with more file requests involved in the system, which injects more available bandwidth resources to the system. Meanwhile, the percentage of replicas hold by NAT peers decreases, meaning that more replicas can be retrieved from direct-connect peers. This helps to achieve more efficient peer assistance, thus cut down on the server capacity cost.

In summary, server contribution, especially the peak server stress, is strongly correlated with the percentage of NAT peers. Since a majority of peers are behind NAT, more advanced NAT traversal techniques are desired in future implementation to exploit the potentially huge amount of bandwidth resources among NAT peers, hence reduce the server cost. Another important implication is that future design of server strategy needs to dedicatedly take into account the time-dependent behaviors of both NAT and direct-connect peers to avoid prohibitive peak server stress and better match the natural evolution of online storage and content distribution application.

## IV. CONCLUSION

With full knowledge of architecture and protocols of a large-scale peer-assisted online storage system, FS2You, this paper, for the first time, examines the role of servers and critical factors that influence the server contributions. By analyzing a large set of real-world traces that we have collected, we demonstrate that the system is able to scale to a large number of users while conserving substantial server capacity. More importantly, we show that server contribution, especially the peak server stress, is strongly correlated with file popularity, time evolution, and the percentage of peers behind NAT involving in the system. By discussing the in-depth rationales behind the findings, we expose the flaws in current design, which we believe of great importance for future design.

## REFERENCES

[1] [Online]. Available: http://www.rayfile.com

[2] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, “Measurements, Analysis, and Modeling of BitTorrent-like Systems,” in *Proc. of ACM Internet Measurement Conference (IMC 2006)*, Oct. 2005.

[3] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, “The Bittorrent P2P File-sharing System: Measurements and Analysis,” in *Proc. of International Workshop on Peer-to-Peer Systems (IPTPS)*, Feb. 2005.

[4] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, “Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload,” in *Proc. of ACM Symposium on Operating Systems Principles*, Oct. 2003.

[5] D. Stutzbach and R. Rejaie, “Understanding Churn in Peer-to-Peer Networks,” in *Proc. of ACM Internet Measurement Conference (IMC 2006)*, Oct. 2006.

[6] S. Das, S. Tewari, and L. Kleinrock, “The Case for Servers in a Peer-to-Peer World,” in *Proc. of IEEE International Conference on Communications (ICC 2006)*, Jun. 2006.

[7] C. Wu, B. Li, and S. Zhao, “Multi-Channel Live P2P Streaming: Refocusing on Servers,” in *Proc. of IEEE INFOCOM 2008*, Apr. 2008.

[8] B. Li, S. Xie, Y. Qu, Y. Keung, C. Lin, J. Liu, and X. Zhang, “Inside the New Coolstreaming: Principles, Measurements and Performance Implications,” in *Proc. of IEEE INFOCOM 2008*, Apr. 2008.