FS2You: Peer-Assisted Semipersistent Online Hosting at a Large Scale

Fangming Liu, *Student Member*, *IEEE*, Ye Sun, Bo Li, *Senior Member*, *IEEE*, Baochun Li, *Senior Member*, *IEEE*, and Xinyan Zhang

Abstract—It has been widely acknowledged that online file hosting systems within the "cloud" of the Internet have provided valuable services to end users who wish to share files of any size. Such online hosting services are typically provided by dedicated servers, either in content distribution networks (CDNs) or large data centers. Server bandwidth costs, however, are prohibitive in these cases, especially when serving large volumes of files to a large number of users. Though it seems intuitive to take advantage of peer upload bandwidth to mitigate such server bandwidth costs in a complementary fashion, it is not trivial to design and fine-tune important aspects of such peer-assisted online hosting in a real-world large-scale deployment. This paper presents *FS2You*, a large-scale and real-world online file hosting system with peer assistance and semipersistent file availability. *FS2You* is designed to dramatically mitigate server bandwidth costs. In this paper, we show a number of key challenges involved in such a design objective, our architectural and protocol design in response to these challenges, as well as an extensive measurement study at a large scale to demonstrate the effectiveness of our design, using real-world traces that we have collected. To our knowledge, this paper represents the first attempt to design, implement, and evaluate a new peer-assisted semipersistent online file hosting system at a realistic scale. Since the launch of FS2You, it has quickly become one of the most popular online file hosting systems in mainland China, and a favorite in many online forums across the country.

Index Terms—Online hosting, content distribution, peer-to-peer protocols, measurement studies.

1 Introduction

Online file hosting systems serve one simple but fundamental purpose: they allow end users to upload files, of both small and large sizes, to the "cloud" of the Internet, to be shared among a group of interested users. As online file hosting systems evolve, they have become increasingly popular and intuitive to use, and the current generation is alternatively referred to as *one-click hosting* services. These services allow end users to upload files onto dedicated servers provided by the hosting service, mostly free of charge. Most such services simply return a URL that can be shared to others (e.g., in discussion forums), 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 peer-to-peer (P2P) file sharing services of the previous generation, such as BitTorrent [1].

As online file hosting systems become increasingly popular, however, server bandwidth costs have become prohibitively expensive, as files are hosted in either content distribution networks or dedicated large data centers. *Rapidshare* [2], one of the most well-known one-click hosting systems, deployed a total of 1,500 terabytes of online storage in its data centers in Asia alone. Skyrocketing bandwidth costs from server-based architectures have made it necessary for all online file hosting systems to impose certain restrictions so that they can afford to remain free of charge to users. These restrictions include download bandwidth limits per day, file size limitations, as well as a maximum time period that files may remain available online.

Though it may seem intuitive to take advantage of peer bandwidth contributions to mitigate server bandwidth costs, the architectural and protocol design of such a peer-assisted online file hosting system should not be taken lightly. It is nontrivial to design and fine-tune a new system that utilizes peer bandwidth contributions in a complementary and transparent fashion, without sacrificing the ease of use, reliability, and performance of one-click hosting services. The architectural design should be able to scale to a large number of users, and to withstand the test of real-world usage over a long period of time.

In this paper, we present FS2You [3], [4], a real-world online file hosting system that we have designed, implemented, and deployed to provide one-click hosting services *with peer bandwidth assistance*. FS2You is designed to dramatically mitigate server bandwidth costs, while maintaining the ease of use and performance comparable to the best server-based solutions. In response to a number of fundamental challenges, we present the architectural and protocol design in our system, and carry out an extensive measurement study to evaluate its performance, based on real-world traces involving millions of users over a long period of time. Since the launch of FS2You, it has quickly

- *F. Liu, Y. Sun, and B. Li are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon 999077, Hong Kong. E-mail: {lfxad, yesun, bli}@cse.ust.hk.*
- *B. Li is with the Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, ON M5S 3G4, Canada. E-mail: bli@eecg.toronto.edu.*
- *X. Zhang is with the Roxbeam Media Network Corporation, No. 52, West North 4th Ring Road, Haidian District, Beijing 100080, China. E-mail: tom.zhang@roxbeam.com.*

Manuscript received 30 Mar. 2009; revised 12 Sept. 2009; accepted 6 Jan. 2010; published online 26 Jan. 2010.
Recommended for acceptance by H. Jiang.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2009-03-0142.
Digital Object Identifier no. 10.1109/TPDS.2010.28.

become one of the most popular online file hosting systems in mainland China. We describe detailed design elements in this system, and analyze the reasons motivating its performance benefits and popularity.

Rather than a research prototype or a simulation based study, we design, realize, and fine-tune a fully working system at a large scale over the Internet, which offers original contributions to the design of a new generation of peer-assisted online file hosting applications. Rather than a pure measurement study largely based on "black-box" approaches, which typically rely on passive techniques such as sniffing to peek into a real working system in trying to interpret external behaviors, our complete knowledge of all internal mechanisms is able to not only validate our design but also help to pinpoint potential design inefficiencies. Furthermore, while many existing measurement studies have solely focused on peer behavior for the lack of knowledge on server policies, our design and measurements have investigated both peer-side and server-side aspects in a cohesive manner. We believe that this offers a rich set of observations and understanding on server-side design in such a new application scenario, which are of great importance for future work on system designs.

To the best of our knowledge, this paper represents the first attempt in the literature to design, implement, deploy, and evaluate a real-world peer-assisted online file hosting system, supported by large volumes of measurement traces. We are confident that this work is of substantial value toward an in-depth understanding of how peer bandwidth contributions and strategic server resource provisioning may be utilized in a complementary and transparent manner.

The remainder of this paper is organized as follows: In Section 2, we highlight our contributions in the context of related work. In Section 3, we present the architectural and protocol design of FS2You, a real-world online file hosting system that we have designed and implemented. In Section 4, our instrumentation and measurement methodologies are described. In Section 5, we show our large-scale trace-driven measurement studies to evaluate important aspects of FS2You. Finally, we conclude the paper in Section 6.

2 Peer-Assisted Online Hosting: A Different Game

2.1 Differences from Other Content Distribution Applications

1. Although peer-to-peer (P2P) file sharing systems have received significant research attention, there exist several important differences between P2P file sharing and peer-assisted online hosting systems.[1] P2P file sharing systems do not use servers to store actual file content, as all files are exchanged among users. As a result, they have no guarantees on file availability, and files being downloaded may become unavailable at any time when all "seeds" (peers with a complete copy of the file) leave the system [5]. In contrast, peer-assisted online hosting systems couple peer assistance and strategic server provisioning in a complementary manner, in order to provide high file availability and downloading performance. As we will show in our measurement studies, such a design philosophy that we have followed in FS2You turns out to be more important for peer-assisted online hosting systems wherein a large number of files are being served with highly diverse popularity, with less popular ones (with fewer peers involved) combined constitute more than a negligible portion of user demands.

2. Online hosting and live video streaming are two different applications with different focuses, design, and issues. First, while live video streaming requires timely and sustained streaming delivery to ensure all the participating peers can receive the *live* video content at the playback rate, online hosting systems focus on relatively large transfers that are less sensitive to the delays for real-time data streaming within a small moving playback buffer. Second, FS2You, as an online file hosting system, focused on the allocation of limited storage to potentially enormous amount of data files so that semipersistent file availability may be achieved. This is not an issue in video streaming systems, such as UUSee [6] and LiveSky [7]. Last but not the least, the usage pattern of peer-assisted online hosting systems, with respect to one-click uploading, storing, and sharing of files, is apparently different from that of live video streaming systems for broadcasting live video programs.

3. Video-on-Demand (VoD) is also different from online hosting services. VoD services allow peers to watch different parts of a video at the same time, and enable user interactive functions, such as random seeks to an arbitrary playback point with stringent delay requirement. In particular, the key design objective of a VoD system is to guarantee continuous playback and short buffering delays after a random seek or an initial startup, which is not an issue in online hosting systems for conveniently uploading, storing, and sharing files. As we will discuss in Section 3.1, the design objective of peer-assisted online hosting systems is to maintain a semipersistent nature of file availability and improve the downloading performance, while conserving valuable server bandwidth and storage costs by taking advantage of peer-assistance. Due to their different focuses, objectives, and usage patterns, the actual design and implementation of VoD systems (e.g., PPLive [8]) is different from that of peer-assisted online hosting systems (e.g., FS2You), even though both can leverage the general idea of hybrid CDN+P2P.

4. There are vital differences between peer-assisted online hosting systems and pure CDN-based solutions. First, peer-assisted online hosting systems, mostly offering free of charge services, have no Service-Level Agreement (SLA) between the service provider and the user, which on the other hand is typically required in CDNs. In particular, files in peer-assisted online hosting systems will be maintained on a semipersistent basis due to the limits of storage space. Due to this lack of SLA and guarantees, the service provider can explore a design space to remove content whenever necessary; however, the key challenge is how content can be removed with minimum degradation of user experience. As evidenced by their increasing popularity and traffic in the Internet, there is *indeed* a market for such online hosting systems (even without service guarantees), due to its convenience to pass a file, large or small,

1. For the sake of brevity, online file hosting systems are henceforth referred to as *online hosting systems* in the remainder of this paper.

from one user to the next without the need for paying for premium storage space.

Second, while pure CDN based online hosting systems solely rely on dedicated servers with prohibitive server bandwidth costs, peer-assisted online hosting systems are designed to take advantage of peer bandwidth contributions to mitigate such server bandwidth costs in a complementary and transparent manner, without sacrificing the ease of use, reliability, and performance of one-click hosting services. In particular, our design and implementation of FS2You have substantially saved server bandwidth costs, while still maintaining a high level of service availability and favorable downloading performance, as evidenced by our measurement results in Sections 5.1.1, 5.2.2, 5.2.3, and 5.4.1.

2.2 Related Work on Hybrid CDN+P2P

In principle, the architecture of FS2You can be categorized within the spectrum of hybrid CDN+P2P solutions, whose potential benefits have been hypothesized in a number of research works in the literature (e.g., [9]). However, most of these prior works on hybrid CDN+P2P are still restricted to concept reasoning (e.g., [10]), theoretical analysis (e.g., [11]), and simulation-based studies (e.g., [9]), instead of concrete and convincing practice in the real world to validate the promise of such an approach. In contrast, our work in FS2You has practically applied such an approach and fully realized its potential in a new real-world system at a large scale over the public Internet. We are not aware of any published study fully describing any working hybrid system in the new application scenario of online hosting at such a large scale.

2.3 Comparison with Other Measurement Studies

While there exists a large number of measurement studies on understanding file characteristics, user dynamics, and downloading performance of existing P2P file sharing systems (e.g., [5], [12], [13]), to date, we are only aware of one measurement study of the emerging new type of online hosting and sharing service [14]. They used a "black-box" approach, such as limited scale of passive monitoring, to infer content characteristics, usage patterns, and service architecture of a pure server-based online hosting system. In sharp contrast, our work used a "white-box" approach with complete knowledge of all internal mechanisms, and the measurements were conducted at a much larger scale. In particular, our work has provided unique measurement results with respect to the effects of the collaboration between peers and servers on the service quality of peer-assisted online hosting systems (e.g., Section 5.4.1), which help to pinpoint potential design inefficiencies.

There are other "white-box" measurement studies that have offered valuable understanding and practical experiences on video streaming quality, user behavior, and server load of P2P live video streaming (e.g., UUSee [6], LiveSky [7]), and VoD systems (e.g., PPLive [8]). However, their focus, design, and issues are different from that of peer-assisted online hosting systems, as we have discussed in Section 2.1. In particular, our measurements have provided unique insights into the effects of server storage allocation across files of highly diverse sizes and popularity, in order to maintain semipersistent file availability (e.g., Sections 5.1.3 and 5.2.3), which is not an issue in video streaming applications. We hope that our work, as the first reported study of various aspects of a real-world peer-assisted online hosting system, can also serve a similar role to guide the development of such a new type of application.

More recently, research attention in measurement studies has also focused on YouTube [15], [16], a popular server-based on-demand streaming system, including its usage patterns, file characteristics, as well as distribution of requests across videos. Although one can consider YouTube and other similar platforms as online hosting solutions specifically designed for multimedia, they solely rely on dedicated server bandwidth and are considered to be server-based solutions.

Overall, this paper offers original contributions that are substantially different from the existing literature. Granted, a large portion of our work focuses on measurement studies on FS2You. For example, we analyze typical performance metrics and user behavior, and examine the correlation among file popularity, user requests, file sizes, and the efficiency of peer assistance. That said, rather than pure measurement studies that treat real-world systems as "black boxes," we first clearly identify our design objectives and challenges, and then propose our solutions that are custom tailored to the challenges identified. As a result, FS2You, a peer-assisted online hosting system, has been implemented and deployed in the real world at a large scale, with extensive trace-driven measurement studies. Such a "closed-loop" research methodology has not been previously applied to peer-assisted online hosting systems, and is rarely seen in the literature on measuring server-based online hosting and P2P file sharing systems as well.

3 FS2You: Challenges and Design

In this section, we first identify the major challenges as we design FS2You, a peer-assisted semipersistent online hosting system. In response to these challenges, we present the system architecture and main components of our design, including the management of peer topologies, the design of peer assistance protocols, as well as server-side strategies.

3.1 Design Objective and Challenges

Two extremes of the cost-performance tradeoff exist in the design of file sharing systems. P2P file sharing systems provide no guarantees on file availability, while server-based online hosting systems are able to provide such guarantees, at the prohibitive cost of server bandwidth and storage. The design objective of a peer-assisted semipersistent online hosting system is to achieve a *reasonable and balanced tradeoff* between these extremes, as we conserve valuable bandwidth and storage resources on servers by taking advantage of peer assistance, while still maintaining a semipersistent nature of file availability, as well as improving the downloading performance. To achieve such an objective, the following challenges need to be addressed:

- The reduction of server involvement may bring adverse effects on file availability and downloading performance. *How do we substantially conserve server bandwidth costs, while mitigating such adverse effects and maintaining an adequate level of service quality and user experience?*

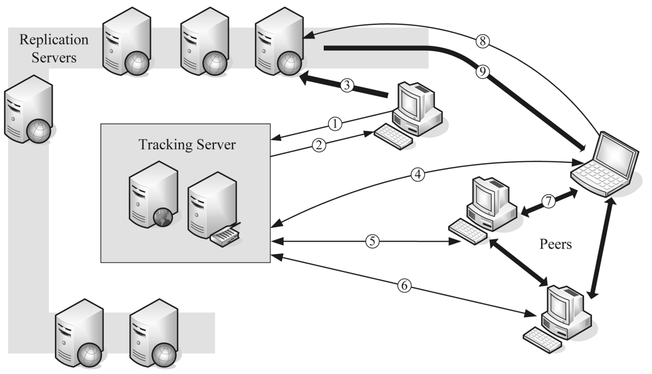


Fig. 1. The architectural design and main components in the FS2You online hosting system. Arrows 1, 2, and 3 represent the interaction between a participating peer and dedicated tracking and replication servers for uploading new content. Arrows 4, 5, and 6 represent the interaction between peers and the tracking server to maintain the peer topology. Arrow 7 represents the sharing of file blocks and exchange of availability among peers. Arrows 8 and 9 represent peer requests and server responses, when the requests cannot be satisfied by other peers alone.

- As the system scales up to a large population, we intend to store contents that are as valuable to users as possible, with a limited amount of server storage space. While recognizing that files will be available on a *semipersistent* basis, *how do we mitigate the drawback of degraded file availability with a limited pool of server storage?*

3.2 Architectural Design

To address the aforementioned challenges, Fig. 1 illustrates the design of the overall system architecture and interactions among main components in the *FS2You* design. In FS2You, each file provided by users is treated as a *channel*, since it is distributed live on an ongoing basis to other users. Each end user is treated as a *peer*, inheriting the terminology of pure P2P systems. The architectural design of FS2You involves a number of main components:

- The *tracking server* serves the purpose of maintaining channel information and bootstrapping peers. Specifically, it maintains a unique *channel ID* and a secure hash value (computed using MD5) for each file provided by a peer; and keeps track of the group of users that are participating in the distribution of each channel.
- *Replication servers* serve as dedicated content servers to maintain availability of channels when they are not actively served by peers alone. There are 60 dedicated replication servers in our implementation of FS2You in China.

As in other P2P file sharing systems, there are two types of peers in the FS2You architectural design. Those who upload files to servers (referred to as *uploading peers*), and those who download only (referred to as *downloading peers*). Uploading peers interact with the tracking servers to create a channel for each file, with its secure hash value and a unique URL for subsequent downloading purposes.

3.3 Managing Peer Topologies

All peers involved in a channel, i.e., either downloading the file in the channel or holding a replica of the file, are organized into a *topology*, so that block availability information can be exchanged and blocks can be shared among neighboring peers (who are *partners* to one another) in the topology. As we will show in our measurement studies, peers in FS2You are highly dynamic and less popular files constitute a large portion of the downloading demand. This observation makes it nontrivial to utilize the maximum amount of peer upload bandwidth, so that server bandwidth costs can be mitigated. How should we construct and manage peer topologies for all channels by judiciously selecting partners for peers, so that peer resources can be utilized as fully as possible?

As our design choice, FS2You combines coarse-grained tracking servers and decentralized gossip protocols for constructing and managing peer topologies. When a downloading peer joins FS2You, it contacts the tracking server and obtains a list of initial *partners* (determined by a system parameter M as will be discussed in Section 4), that are randomly selected from peers associated with the same channel. As we shall further elaborate below, this partnership list is periodically updated and new partners can be added. Partners can be *active* or *inactive*, which is determined by whether there are actual connections and data block transfers between the peer and its partners.

Peers need to keep a reasonably number of active and inactive partners in order to maintain a sustainable level of downloading efficiency and to be resilient to network dynamics. In FS2You, each peer can have up to N partners in its pool of *inactive* partners, which is related to the number of peers simultaneously being online in a channel, as we shall explain in Section 4. In case the size of the inactive partner pool increases to over the threshold N, a peer will discard *aged* partners or partners who it has failed to establish a connection with. On the other hand, with respect to *active* partners, the maximum number per channel that each peer could have is bounded by a system parameter K_{max} in FS2You. Connections to active partners can be broken from time to time due to various reasons, such as slow downloading rates and idling for a long time. Each peer periodically monitors the number of its current active partners. If the number of active partners falls below a threshold K_{min}, it exchanges active partner lists with its current active partners through decentralized gossip [17] and attempts to establish new connections with a random subset of inactive partners; if successful, the status of these inactive partners will be promoted to active. In practice, the setting of the maximum and minimum number of active partners is similar to that of BitTorrent, as will be configured in Section 4.

How do we maintain accurate lists of peers in each channel on the tracking server? In FS2You, peers report their status to the tracking server periodically (in a coarse-grained period T_s to reduce the burden on the tracking server, as will be configured in Section 4), which contains vital peer information such as a unique peer identifier, its IP address, and information about channels that it has joined. Since each peer can be potentially involved in a large number of channels, to keep overhead low, the reported status only includes information of a bounded number of top C channels, including the channel identifiers and *download ratios*, defined as the amount of file that has been downloaded so far. The top C channels represent files that have been downloaded by this peer, ranked by a value computed by the combination of the *file size*, *download ratio*, and *the time when the peer joined the channel*. Intuitively, the larger the file size or the download ratio is, the higher the ranking is. In addition, channels that the peer joined later gain higher ranking values. 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 downloading performance can be improved, and the load on servers can be alleviated.

3.4 Content Distribution Strategies

Each file is divided into fixed size blocks. A *Block Map* (BM) is introduced to specify the availability of blocks at each peer [17]. The periodic exchange of BMs among peers enables them to locate the needed blocks. Each peer can retrieve a number of distinct blocks from multiple active partners simultaneously up to the number of its current active partners. If multiple partners hold a desired block, a peer will randomly choose one of its active partners to request that block.

FS2You implements a unique *sequential block scheduling* mechanism as follows:

- The first block is always fetched from the server. Intuitively, this reduces the latency for obtaining the first block, and improves the user downloading experience.
- Block scheduling is periodically scheduled, with a period of time T_b. The selection of this parameter needs to balance between signaling overhead and peer upload bandwidth utilization. A short period incurs extra signaling overhead, while a long period could potentially underutilize peer upload bandwidth. Generally, given a block size B and the upload capacity of a peer u_p, the block scheduling period can be tuned as $T_b = t_b + \Delta t$, where $t_b = B/u_p$ is the estimated minimum time for transmitting one block, and Δt is used to shift the scheduling period from aggressive to conservative. The corresponding customization in FS2You will be presented in Section 4. In each round of block scheduling, a peer sequentially requests missing blocks up to the number of its current active partners. The use of such sequential block scheduling also takes account of the purpose to upgrade the system to support Video-on-Demand (VoD) service, by adding a media player at client (peer) side.
- To improve file availability and the downloading experience, peers are allowed to request help from replication servers, but only when any of the following three conditions holds, in order to prevent server bandwidth abuse: 1) There are currently no active partners, e.g., the file is unpopular or a peer fails to establish connections with any of its partners; 2) None of the active partners hold the desired block; 3) The aggregate downloading rate from active partners (i.e., the size of data that was downloaded from active partners over the previous scheduling period) falls below a *request-from-server* threshold D. This threshold aims to prevent peers from aggressively consuming server bandwidth, as will be empirically determined in FS2You in Section 4.

3.5 Server Strategies

Replication servers in FS2You not only provide online storage, but also cooperate with content distribution. There are three sets of strategies that servers adopt in the design of FS2You to facilitate storage and content distribution, tailored for *uploading*, *downloading*, *hosting* services, respectively.

▹ *Uploading service.* In FS2You, users are allowed to upload a variety of files to servers without *any* size or format limitations. This policy attracts millions of users to upload a huge volume of content to FS2You, catapulting it to one of the most popular online hosting systems in China in a short period of time. Our measurements have shown that 500 GB to 1 TB of files are routinely uploaded each day. To cope with such a demand without consuming excessive server resources, the following two strategies are adopted: 1) When a user requests to upload a file, the system ensures that only one copy is stored in one of the replication servers; and 2) this copy is stored in the server nearest to the user requesting the upload. This helps to reduce the uploading time, and to mitigate unnecessary inter-AS traffic.

▹ *Downloading service.* Servers complement peers to supply file blocks, especially to those peers suffering poor downloading rates, e.g., below the request-from-server threshold D. The challenge, however, is how to properly satisfy a potentially large number of requests without incurring prohibitively high bandwidth costs. In FS2You, in accordance with our content distribution strategies, when a server receives a block request, it makes its decisions based on the following policies: 1) If the request is for the first block of a file, it will be served immediately. 2) The request for other blocks 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 policy is that, a larger number of references will likely result in more copies of the file shared among peers, implying that the servers should serve less. This simple policy implicitly allows peers involved in popular channels to largely rely on peer assistance rather than servers, and allocate more server resources to unpopular files with fewer peers. In our forthcoming measurement studies in Section 5, we will examine how the strategy influences file downloading rates and user experience with different popularity levels.

▹ *Hosting service.* With a limited pool of server storage space, it is not feasible to host unlimited volumes of files on replication servers, and files in FS2You can only be *semipersistent* in nature. We wish to mitigate the drawback of such a semipersistent nature of file availability, and to maintain a high level of user satisfaction. The rule of thumb we have followed in FS2You is to maintain the availability of recently or frequently accessed files, while replacing less popular ones when necessary. Files with a reasonable level

TABLE 1
System Parameters of FS2You

M	Number of initial partners	20
N	Maximum size of the inactive partner pool	500
K_{max}	Maximum number of active partners per channel	32
K_{min}	Minimum number of active partners per channel	16
T_s	Peer status report period (minutes)	5
C	Number of top channels included in status report	20
T_b	Block scheduling period (seconds)	5
D	Peer request-from-server threshold (KB/second)	10
T_l	Peer snapshot report period (hour)	1
h	File reference index threshold (MB/hits)	100

of user demand remain available as long as they do not impose an overwhelming load on servers.

In particular, the following strategies are used in FS2You: 1) Small files (with a size below 10 MB) will not be deleted unless specifically demanded by the original uploading user. 2) Each file i is assigned a *reference index* H_i, which monitors the ratio between uploaded file sizes and file access frequencies. More specifically, let S_i be the size of file i, and F_i be its daily access frequency (i.e., a daily tally of unique IP addresses that have accessed file i). The reference index H_i is calculated per day as $H_i = S_i/F_i$. In FS2You, if H_i is lower than a particular threshold h (will be empirically configured in Section 4.1 according to our real-world measurement results in Section 5.1.3), the file is either small or frequently accessed, and as such should remain persistent in the servers. On the other hand, if H_i is higher than the threshold h for a sustained period of time (set to five days in FS2You), the file will no longer be hosted on the servers. The rationale is to store large files only if substantial user interests and popularity persist, in order to avoid excessive use of server storage.

4 FS2You: System Configuration and Collection of Traces

In order to validate the effectiveness of our architectural design, we have implemented FS2You and made it readily available for users (mostly in mainland China) to use. It has quickly become one of the most popular online hosting services in China after its deployment. In this section, we first describe the practical system configuration of FS2You. Then, to evaluate and analyze the performance of FS2You, we have implemented a detailed instrumentation mechanism, which helps collect a large volume of real-world traces.

4.1 System Configuration

To provide better reliability, FS2You deployed a set of four dedicated tracking servers, which were strategically placed in different domains. The requests from peers can be routed to one of the tracking servers that are resolved by DNS. For scalability, by benefiting from the gossip mechanism described in Section 3.3, FS2You system is able to choose a coarse-grained period for peer status report ($T_s = 5$ minutes) to reduce the burden on the tracking servers, which is practically adopted in modern real-world peer-assisted systems [18]. A lower reporting frequency can further offload the tracking servers, but at the expense of less fresh peer lists for channels.

Table 1 summarizes the system parameters described in Section 3, with their corresponding configuration in FS2You based on well-established reference values and concrete evidence from our real-world measurement results.

More specifically,

1. the number of initial partners for a peer is set as $M = 20$, which is a reasonable setting for typical gossip protocols adopted in real-world peer-assisted systems, such as Coolstreaming [17]. Practically, as long as this number is not too small, it also worked well in FS2You system.
2. According to observations in our measurements, the number of peers simultaneously being online in a popular channel could be in the order of hundreds; and hence FS2You allows the size of the inactive partner pool for a peer to be up to $N = 500$, which is expected to cover most, if not all, of the peers in the same channel.
3. To keep a reasonably number of active partners while limiting the peer management overhead, our experience in FS2You suggests that the sweet spot for the minimum and maximum number of active partners per channel for a peer is $K_{min} = 16$ and $K_{max} = 32$, respectively, which is also compatible with other practical peer-assisted systems, such as BitTorrent [1] and PPLive [8].
4. While FS2You is specifically designed for use in mainland China, where DSL peers rarely enjoy upload capacities exceeding 512 Kbps, given a typical block size of 256 KB [1], the estimated minimum time for transmitting one block is 256 KB / $512\ \text{Kbps} = 4$ seconds. Without being over aggressive, a conservative value of $T_b = 5$ seconds is chosen as the block scheduling period (Section 3.4).
5. Empirically, FS2You sets the request-from-server threshold as $D = 10$ KB/second to prevent peers from aggressively consuming server bandwidth, whose effects will be further discussed in Section 5.4.
6. Finally, the file reference index threshold used for the hosting service (Section 3.5) is empirically set to $h = 100$ MB/hits, which will be shown to be effective in Section 5.1.3.

4.2 Collection of Traces

Each peer in FS2You is designed to report its activities and status to the trace server,[2] using the HTTP protocol. The trace server appends the time of receipt to each report, and then stores it locally in log files, with a maximum size of 64 MB in each file. Traces on the order of hundreds of Gigabytes are collected every month. For example, 350 GB traces have been collected by the trace server from 3.3 million FS2You peers, over a one-month period from June 21 to July 18, 2008.

In our subsequent measurement studies, we focus on analyzing two types of reports: *Download Event Summary* (henceforth referred to as *"summary"*) and *File Source Snapshot* (henceforth referred to as *"snapshot"*).

2. Generally, the trace server can be either dedicated end host(s) or as part of the functions of the tracking server. During our measurement period, we use the tracking servers as the trace server.

TABLE 2
Snapshots: An Example as Reported by One of the Peers

Time of reporting	Peer ID	Channel ID	Download ratio
t_1	P_1	F_1	100%
		F_2	50%

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

The *snapshot* records statistics about files that a peer contributes, and is reported periodically (with $T_l = 1$ hour).[3] The critical information in the *snapshot* contains the reception time of the snapshot at the trace server and the download ratio. As we stated in Section 3, the download ratio represents the percentage that the file has been downloaded by the reporting peer so far. Table 2 shows an example snapshot, in which a peer with peer ID P_1 reports a snapshot to the trace server at time t_1. At the time of reporting, P_1 locally stores two files with channel IDs F_1 and F_2, and their download ratios are 100 percent and 50 percent, respectively. This indicates that P_1 holds a complete replica of F_1 and 50 percent of the blocks in F_2, both of which can be served to other peers.

It is inevitable that there could be inaccuracies in our instrumentation and trace collection mechanism, due to clock skew, crash failures of peers and the trace server. For example, the trace server had indeed suffered from a crash failure, and the service was interrupted on three days in July. The coarse granularity of reporting snapshots (once per hour) is designed to reduce reporting overhead as the system scales up, but it also introduces a degree of inaccuracy when it comes to estimating the period of time that a peer remains online. We are convinced that the large volume of traces that we have collected is valuable even with such imperfections due to real-world complications, as we shall demonstrate in the next section.

5 MEASUREMENT RESULTS AND ANALYSIS

To extensively evaluate the architectural and design choices of FS2You, we now take advantage of the large volume of traces we have collected from over three million real-world users, and analyze our measurement results to study a number of important aspects of FS2You, including the overall scale, file availability, server bandwidth costs, and the level of service quality. Our measurement studies will close the "loop" of our research methodology, and will validate the effectiveness of our design.

3. Note that the snapshot report with a period of T_l is different from the status report with a period of T_s. The former is only used for trace collection and measurement purposes, rather than for operational reasons, while the latter is used for maintaining fresh lists of peers in each channel on the tracking server, as elaborated in Section 3.3.

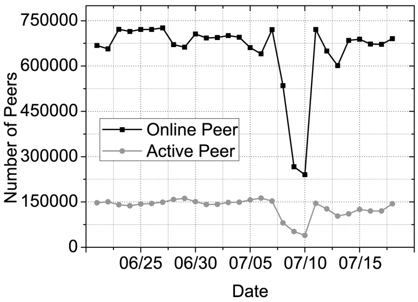


Fig. 2. The number of FS2You online peers and active peers from 21 June to 18 July 2008. The sharp decrease on 8, 9, and 10 July was due to the crash failures of the trace server.

5.1 Overall Statistics

5.1.1 Overall Scale and Performance

To demonstrate the system scale and overall performance, we first present some statistics of FS2You online peers, as well as the evolution of traffic. In FS2You, online peers can be classified into two categories based on their activity: *active* peers with download activities, and *inactive* peers that stay online without download activities. For example, if a peer downloads a file on 1 July, then we say that the peer is active on 1 July; conversely, if a peer stays online on 1 July without issuing any download requests, we say that the peer is inactive on 1 July. Among the total 3,384,948 online peers captured from 21 June to 18 July 2008, 2,240,517 peers were active for at least one day, whereas the remaining 1,144,431 peers were inactive during the entire month. Fig. 2 shows the large number of online peers and active peers over the month. The sharp drop on 8, 9, and 10 July is due to the crash failures of the trace server. Interestingly, we found a "weekend pattern" in that over the weekends (e.g., 21-22 June, 28-29 June, and 5-6 July), the number of online peers regularly decreased (e.g., drops from above 700,000 during weekdays to below 675,000) while the number of active peers remained stable and even slightly increased. We believe that this is because fewer yet relatively more active peers tend to stay online during weekends. This is further evidenced by the traffic evolution in Fig. 3 as discussed below.

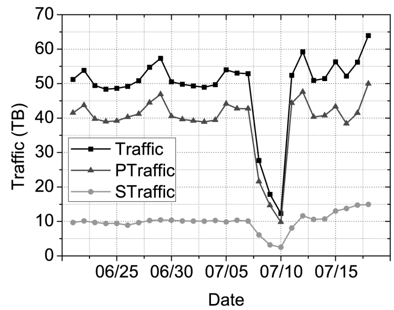


Fig. 3. The total traffic (Traffic), P2P traffic (PTraffic), and server traffic (STraffic) of FS2You from 21 June to 18 July 2008. The sharp decrease on 8, 9, and 10 July is due to the crash failures of the trace server.

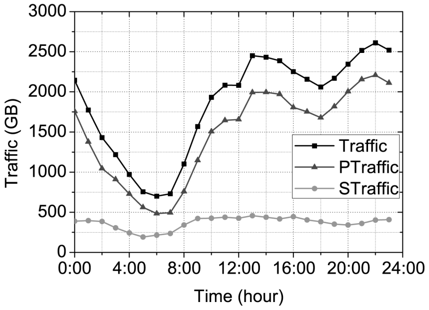


Fig. 4. The evolution of total traffic (Traffic), P2P traffic (PTraffic), and server traffic (STraffic) of FS2You over time on 21 June 2008.

Fig. 3 shows the volume of observed traffic over the month, where *STraffic* stands for the traffic served by replication servers, *PTraffic* represents the traffic contributed by peers, and *Traffic* is the sum of *STraffic* and *PTraffic*. Again, the sharp decrease is attributed to the missing traces. We have made the following observations: 1) The total volume of traffic in the system varied from 49 TB to 65 TB during the month and also showed a weekend pattern: 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 the aforementioned observation that there were relatively more active peers with download demand during weekends, thereby leading to higher traffic. 2) Compared to the total volume of traffic, the server traffic was fairly stable and stayed around 10 TB. 3) Over the entire month, up to 80 percent of the traffic was contributed by P2P delivery, which significantly alleviated the load on the servers.

To investigate how the system scales with peer assistance as the number of peers grows, a closer look at the daily traffic evolution of FS2You on a representative day is shown in Fig. 4. From 6 a.m. to 1 p.m., there has been a steady increase of traffic as an increasing number of users joined the system (e.g., from around 100,000 to around 320,000), and the P2P efficiency (defined as *PTraffic*/*Traffic*) has increased from 70 percent to 85 percent. Specifically, even during the "calm" period (6 a.m. to 8 a.m.) with relatively fewer users (e.g., around 100,000 to 150,000), our design of peer assistance can successfully conserve more than 70 percent of the server bandwidth cost. For the remainder of the time, the P2P efficiency steadily stayed around 80 percent and reached its peak of 85.7 percent at 10 p.m. Furthermore, to explore and understand the rationale behind these observations, a fine-grained investigation with respect to various essential factors including peer types, file popularity, and time evolution will be presented in Sections 5.3.2 and 5.3.3, which brings forth complementary insights on how peer assistance and server strategies can help achieve server bandwidth reduction and thus better system scalability.

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 (on the order of terabytes per day) over a long period of time. It is evident that the cost of server bandwidth has been substantially saved by peer assistance, one of the important design objectives of FS2You.

TABLE 3
Statistics of Active and Inactive Peers

		Daily		Weekly	
		Number	Per.	Number	Per.
Active	Total	147,945		744,062	
	Online next day/week	66,392	44.8%	298,105	40%
	Active next day/week	28,284	19.1%	165,900	22.3%
Inactive	Total	518,354		779,614	
	Online next day/week	390,523	75.3%	527,463	67.6%
	Inactive next day/week	368,016	71%	463,716	59.4%

5.1.2 System Dynamics

To obtain a fine-grained understanding of how peer bandwidth and storage contributions can be utilized under inherent user dynamics, we now characterize the online time and file resource distributions of different categories of peers.

First, we observed that over a long period such as one month, active peers consist of a large portion of the entire population. Among the total 3.3 million peers captured in our traces, 66 percent of them were active. However, within a relatively short period such as a single day, Fig. 2 shows that inactive peers seem to dominate the system. For example, among 666,299 peers that had been online on June 21, 78 percent of them were inactive. The remaining 27 days exhibited a similar phenomenon. It is likely that active peers are highly dynamic while inactive peers are relatively more stable. We show this by analyzing the overlap of peers between two adjacent days and two adjacent weeks, respectively. Table 3 shows that on average, 75 percent of inactive peers showed up on the next day and 71 percent of inactive peers would still be inactive; while only 45 percent of active peers appeared on the next day and 20 percent of active peers would still be active. Statistics between adjacent weeks show a similar trend: 70 percent of inactive peers and 40 percent of active peers showed up in the next week.

We further capture the system dynamics from the perspective of *peer departure rates* of n days, defined as the ratio of the number of peers who have stayed online for exactly n days to the number of peers who have stayed online for $\geq n$ days. Our objective is to characterize the relationship between the peer departure rates and the number of online days. Fig. 5 shows the distribution of the number of days peers stay online, as well as the peer

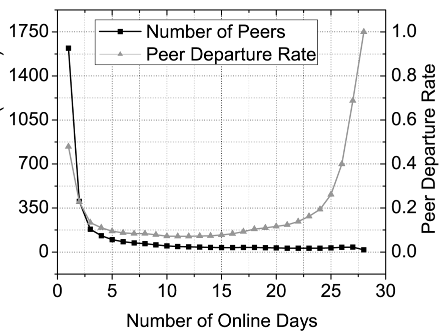


Fig. 5. Distribution of peer online days and departure rate from 21 June to 18 July 2008.

TABLE 4
Comparison between Active and Inactive Peers

Category	Active Peer	Inactive Peer
Online days (day)	4.28	7.63
Online time per day (hour)	5.7	8.7
Files (measured for 4 weeks)	3.86	1.66
Files (measured for 1 day)	5.5	2.4
Contribution	20.2%	79.8%

departure rates. We have observed that up to 47 percent of peers stayed for just one day in the system; this implies that after staying online for one day, 47 percent of peers left the system. This percentage decreases dramatically as the number of online days increases, and stays around 10 percent from 10 days to 24 days. It reveals that short-lived peers, especially those that newly appear in the system, are more likely to leave the system while aged peers (most likely inactive) are relatively stable.

We next compare the time peers stay online and the number of files peers retain, for both active and inactive peers, over a short one-day or a long four-week period. We use the maximum number of files recorded in a peer's snapshot to represent the number of files a peer retains. Table 4 shows that, on average, an active peer in a day retains 5.5 files while an inactive peer retains only 2.4 files. An active peer stays online for around 5.7 hours in one day, while an inactive peer stays online for more than 8.7 hours. When the period is extended to four weeks, an active peer stays online for 4.28 days out of 28 days and retains 3.86 files on average, while an inactive peer stays online for 7.63 days and retains only 1.66 files. It is evident that inactive peers are more stable but retains fewer files, as compared to active peers.

In summary, the large population and stability of inactive peers have the potential to be fully utilized in our design. As we closely compare the upload contribution of peers, we found that up to 79.8 percent of traffic was contributed by inactive ones while 20.2 percent was contributed by active ones. This demonstrates that the available bandwidth resources among peers have been well utilized to improve P2P efficiency, and to alleviate server bandwidth costs.

5.1.3 File Characteristics

We now examine the characteristics of FS2You files, based on the traces from a representative day on June 21, 2008. Among 91,530 diverse requests for a variety of files, we found that around 47 percent of files are compressed archives (e.g., in *rar* or *zip* format), 30 percent are videos, 12 percent are audio, and 11 percent are other types. To our experience, most of the compressed archives are videos. Such skew of file types implies that when using such a system, users are more interested in multimedia content, especially videos.

We further investigate the correlations of file size, file popularity, and file replicas. Fig. 6 plots the average number of file requests and replicas versus file sizes, grouped into different ranges. We have made the following three observations: First, large files (over 300 MB) receive more requests on average than small files (below 300 MB), implying that users generally prefer large files. Specifically, 300 MB to 1 GB is the most popular range, which represents typical sizes of videos.

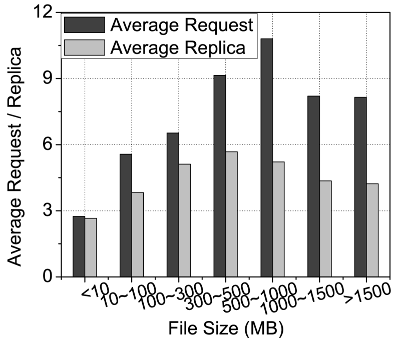


Fig. 6. Average number of file requests and replicas versus file sizes, grouped into different ranges.

Second, the server strategies for the hosting service in FS2You (Section 3.5) effectively guarantee that large files are able to survive in the system only if they are sufficiently popular. On the other hand, to maintain a high level of user satisfaction, a large number of small files can remain available as long as there exists a reasonable level of user demand and they have not occupied excessive storage space. We found that 72.4 percent of observed files have sizes below 100 MB, and they only occupy 21.4 percent of the server storage. These results reflect the design philosophy of semipersistence in an online hosting service to balance the trade-off between file availability (thus user satisfaction) and server storage costs. Finally, the gap between the number of requests and replicas of large files is larger than that of small files. For example, the average number of requests and replicas are nearly the same for the files with sizes below 10 MB, while the gap increases to a ratio of nearly 2 for the files with sizes between 500 MB to 1 GB. This implies that, although users prefer to download large files, they do not tend to keep such large files in their local storage.

In particular, recall that a *reference index* (Section 3.5) is maintained in FS2You to determine whether or not a file would be kept or removed from the online hosting system. It represents the ratio between the size of a file and its daily access frequency. We discovered that the average reference index for files with sizes below 1 GB is lower than a value of 100 MB/hits (e.g., 1 GB / 10); while the average reference index for files with sizes beyond 1 GB is generally higher than 100 MB/hits. This empirical observation results in an reference index of 100 MB/hits as the threshold for removing large yet unpopular files from the system.

5.2 File Availability

5.2.1 File Request Distribution

To explore the file availability of FS2You, we now focus on a representative day (July 2, 2008) with over 570,000 of requests for a variety of files. We first attempt to understand the distribution of peer requests among the observed files. We began with applying the Zipf analysis on the request of FS2You files. Zipf's law states that if objects are ranked by the request count, the popularity of the ith most popular object is proportional to $i^{-\alpha}$, where α is a constant. The Zipf distribution exhibits a linear shape on log-log scale. Fig. 7 plots the file request count versus the descending ordered list of popularity rank on a log-log scale (left y-axis), along

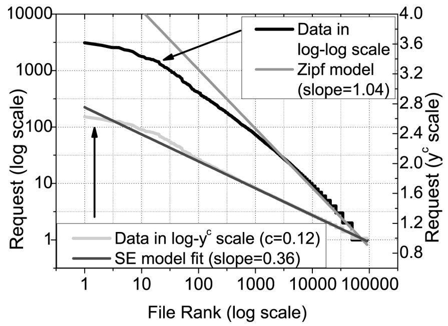


Fig. 7. File request count versus the descending order of file popularity rank on log-log and log-y^c scale, along with linear fit curves. The empirical data fit stretched exponential distribution, rather than Zipf.

with a linear fit curve. We can see that the file request distribution of FS2You does not follow a Zipf distribution. Specifically, the empirical curve is much flatter than the Zipf curve among the most popular files. This implies that the most popular files are significantly less popular than the Zipf prediction. We believe this is caused by the immutability of files, and the "fetch-at-most-once" user behavior [12]. Instead, by plotting the empirical data on log-y^c scale (right y-axis), we found that the FS2You file request distribution can be well fitted with a linear line, indicating that it follows the stretched exponential (SE) distribution [19] with a proper constant $c = 0.12$.

Further, we apply concentration analysis [16] that show how skewed the requests from peers are toward popular files. Fig. 8 plots the cumulative distribution of the file request count and the corresponding traffic, versus the descending order of file popularity rank (normalized). We find that the Pareto principle (80/20 rule) is applicable with respect to both the requests and traffic, and the traffic is even more skewed because popular files usually have relative larger sizes, shown in Fig. 6.

5.2.2 Impact of Peer Assistance on File Availability

We now investigate how peer assistance helps to improve the availability of files. We first examine the number of replicas of files that we have observed in a representative day (July 2, 2008). We found that more than 93 percent of the files have at least one replica among online peers and more than 58 percent of the files have more than 24 replicas reported during that day (more than one replica per hour on average). In terms of bytes, we found that around 12.5 percent of files have a P2P efficiency higher than 75 percent, and these 12.5 percent of files cover 80 percent of the total requests. With these observations, we are able to conclude that peer assistance has led to a substantial improvement with respect to the availability of files in the online hosting system, as more than half of the files are owned by at least one peer at any time during the day, and further requests for these files may not require any bandwidth from the replication servers.

5.2.3 Impact of Servers on File Availability

Switching the perspective to replication servers, we now attempt to study their impact on the availability of files, especially those less popular files. Over a one-week observation from June 21 to June 27, we found that about

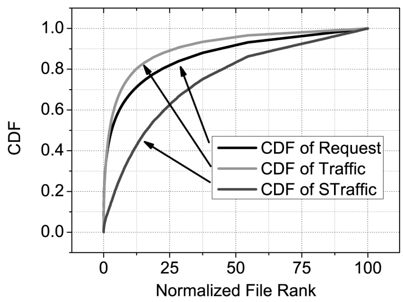


Fig. 8. Cumulative distribution of file request count, the resulting traffic and server traffic versus the descending order of file popularity rank.

80 percent of files are less popular, in that more than half of these files are supplied by servers. Their average request count is 4 while the average request count of all observed files is 14, which also confirms that they are less popular. These files account for about 25.3 percent of the total unique requests observed in the one-week trace. In terms of bytes, these files account for 13 percent of the total system traffic, and 54 percent of the total server traffic. More specifically, about 51 percent of files are completely fetched from servers.

These observations reveal that: 1) Less popular files represent more than a negligible portion of user demand in FS2You, which reflects one aspect of the inherent nature of online hosting services. 2) Due to the lack of partners and replicas, less popular files are usually less likely to benefit from peer assistance. To compensate for the lack of peer assistance, it is necessary for replication servers to provide dedicated storage in support for these files, with the total cost of around 10 TB of storage space (61 percent of the total size of files that appeared in the one-week trace). We further extend our investigation scope to all file requests from peers, and find that up to 95.4 percent of peers successfully received file blocks, and only 4.6 percent of peers failed to retrieve file blocks. This demonstrates a superior level of file availability in FS2You.

5.3 Server Traffic

5.3.1 Overall Server Traffic

To obtain an in-depth understanding of server bandwidth contributions, we made additional observations by revisiting Fig. 8, in which we have plotted the cumulative distribution of server traffic. We have discovered that the CDF corresponding to the server traffic volumes is less skewed than that of both the number of requests and the total traffic volumes, implying that peer assistance effectively mitigated the server load with respect to the most popular files. However, we have also observed that popular files still consumed a larger portion of the total server bandwidth costs, as compared to less popular ones. Specifically, the top 25 percent popular files consists of approximately 62 percent of the total server bandwidth costs.

5.3.2 Categorizing Server Traffic by File Popularity

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 percent of files with

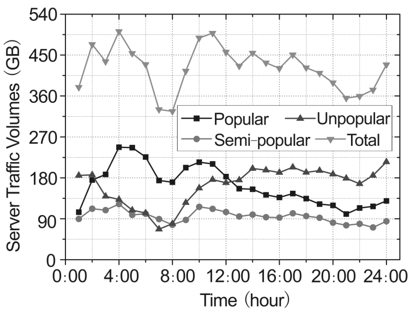


Fig. 9. Total server traffic volumes for three categories of files within a 24-hour period.

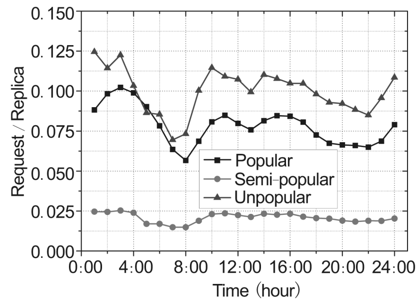


Fig. 10. Request-to-replica ratios of three categories of files within a 24-hour period.

larger number of requests within the measured day; 2) among the remaining 90 percent of files, those with more than 60 replicas (i.e., the average number of replicas of the remaining 90 percent of files) are regarded as *semipopular files*; while the remaining are regarded as *unpopular files*. The semipopular files are often files that were popular in previous measured periods with sufficient replicas among peers.

Fig. 9 plots the total server traffic volumes for three categories of files within a 24-hour period. Fig. 10 plots the *request-to-replica* ratio (defined as the number of requests divided by the number of replicas) of the three categories of files. From this figure, we make the following observations:

First, during a "calm" 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 a significant level of variations with respect to the total sever traffic volumes is attributed to a temporary waiver of the probabilistic serving strategy in the downloading service (Section 3.5) from 2 a.m. to 4 a.m. More importantly, the peak server stress of 500 GB occurs at 4 a.m. as well. These together not only demonstrate the important role of a well-designed server strategy for limiting the server bandwidth costs, but also reveal a natural tradeoff between user experience and server capacity. While the temporary waiver of server strategy attempts to improve file availability and downloading performance, it poses a vital threat to overwhelm servers even during the seemingly safe period with the fewest peers. We will further confirm the risk as we take a closer look at peers behind NAT in the next subsection. In contrast, the total server traffic volumes accounted for by unpopular and semipopular files have both decreased during the "calm" period, as they are rarely requested.

Second, during the transition period (such as from 8 a.m. to 10 a.m.) between "calm" and "active" periods, all the request-to-replica ratios of the three file categories increase significantly, as an increasing number of peers with file requests join the system. The total volume of server traffic has increased for all three file categories, which leads to a later peak of the server load at 11 a.m.

Third, during the "active" period such as from 10 a.m. to 11 p.m., the total server traffic volumes accounted for by popular files decrease remarkably over time, which shows the effectiveness of both peer assistance and the probabilistic serving strategy. On one hand, the *file popularity indices* of popular files become larger as time progresses, which leads to a lower probability to acquire assistance from servers, and thus implicitly directs peers to rely on assistance from other peers. On the other hand, as shown in Fig. 10, the request-to-replica ratio of popular files decreases during the "active" period, which implies that the demand for popular files can indeed be satisfied by peer assistance. In contrast, the total server traffic volumes accounted for by unpopular files remain steadily at a high level of around 200 GB during the "active" period, which dominates the total server bandwidth costs. This reveals that more server resources are allocated to unpopular files with fewer peers; since lower demand levels for a large number of unpopular files have increased the probability for servers to get involved. With respect to semipopular files, the resulting total server traffic volumes stay at a lower level of 75 GB to 100 GB. The rationale is that semipopular files are able to take advantage of a sufficient number of existing replicas to satisfy a smaller number of requests (confirmed by the lower request-to-replica ratio of semi-popular files as shown in Fig. 10), so that the load on servers may be reduced.

In summary, the volume of server contribution is strongly correlated with the file popularity and the time period. The total server traffic volumes for popular files have been effectively reduced during the "active" period by the current design of FS2You.

5.3.3 Categorizing Server Traffic by Peer Types

We next examine the total server traffic volumes from a different perspective, by categorizing server bandwidth consumption to two categories: bandwidth consumed by peers behind NAT (henceforth referred to as *NAT peers*), and bandwidth consumed by directly connected peers with public IP addresses. In the entire set of observed requests, NAT peers make up a dominant subset of 74 percent, which reflects the uneven distribution of FS2You user types. Table 5 compares the traffic (from servers, NAT peers, and directly connected peers) that consumed and contributed by NAT peers and directly connected peers, respectively. We found that: 1) The small portion of directly connected peers contributed 50 percent of traffic, while the large portion of NAT peers only contributed 30 percent 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 percent of the total traffic (39.7 GB) consumed by NAT peers is obtained from servers, which is larger than that of directly connected peers. This implies

TABLE 5
Traffic Statistics of NAT Peers and Directly Connected Peers

	NAT	Directly connected	Total
From Server (TB)	8.25	2.02	10.27
Percentage (%)	21%	16%	20%
From NAT (TB)	10.28	5.72	16.00
Percentage (%)	26%	44%	30%
From directly connected (TB)	21.16	5.23	26.39
Percentage (%)	53%	40%	50%
Total (TB)	39.70	12.98	52.68

that NAT peers are relatively more likely to encounter difficulty in downloading blocks from other peers, thus resort to assistance from servers.

Based on the above observations, we believe that the percentage of NAT peers has a significant impact on the overall supply of bandwidth resources, which in turn can affect the server bandwidth cost. Intuitively, a high percentage of NAT peers with underutilized bandwidth capacities runs a "deficit" in the system by consuming more resources than their contributions. To further show this "deficit" effect, Fig. 11 plots the total server traffic volumes consumed by NAT and directly connected peers, respectively, within a 24-hour period. Fig. 12 plots the percentages of requests, replicas, and total server traffic volumes for NAT peers, within a 24-hour period. The total server traffic volumes consumed by NAT peers make up a dominant portion, and have a larger variation over time than that of directly connected peers. And we have discovered the following:

First, during "calm" period from 2 a.m. to 6 a.m., though the numbers of requests from both types of peers have decreased as shown in Fig. 13, the percentage of requests from NAT peers in Fig. 12 has actually increased substantially. As we have previously discussed, the increasing percentage of NAT peers can bring a negative "deficit" effect to the overall bandwidth supply in the system. For peers suffering a poor downloading performance, this implies that they are likely to request assistance from the 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 demands as much as possible. This explains why the server load has peaked at 4 a.m., and again confirms the potential risk during the "calm" period.

Second, during the transition period from 8 a.m. to 10 a.m., Fig. 13 shows that the amount of requests from both types of peers increases; meanwhile, the percentage of replicas accounted for by NAT peers increases as well and reaches its peak of 85 percent at 10 a.m., as depicted in Fig. 12. This implies that while an increasing number of peers with file requests join the system, a majority of replicas are actually held by NAT peers, and as such are under-utilized to satisfy the demand. This is the reason that leads to the second peak server stress at 11 a.m.

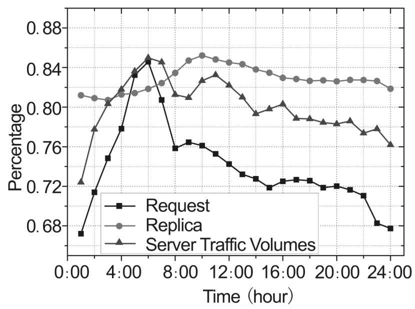


Fig. 12. Percentages of requests, replicas, and total server traffic volumes accounted for by NAT peers, within a 24-hour period.

Third, as shown in Fig. 12, during the "active" period there is a relatively higher percentage of directly connected peers with more file requests involved in the system, which injects more available bandwidth resources to the system. Meanwhile, the percentage of replicas held by NAT peers decreases, meaning that more replicas can be retrieved from directly connected peers. This helps to achieve more efficient peer assistance, and as such reduces the server bandwidth cost.

In summary, we have observed that server bandwidth contributions, 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 a future implementation of FS2You to exploit the potentially large amount of unused bandwidth resources among NAT peers, and to reduce the server bandwidth cost. Another important lesson learned is that server strategies need to meticulously consider the time-varying behavior of both NAT and directly connected peers, so that prohibitive server bandwidth stress at peak times can be avoided.

5.4 Service Quality and User Experience

5.4.1 Overall Service Quality

First, we examine the service quality of FS2You by exploring the correlation between service quality and the level of server involvement. For each observed file, we define: 1) the

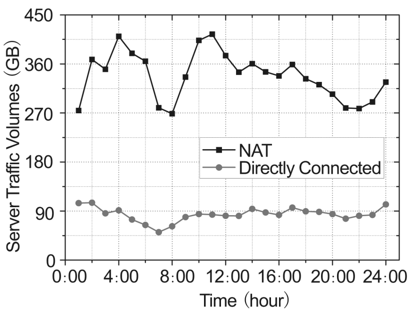


Fig. 11. Total server traffic volumes for NAT peers and directly connected peers within a 24-hour period.

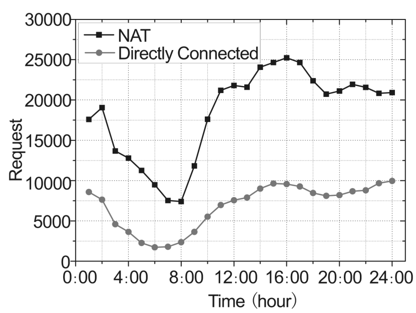


Fig. 13. Number of requests from NAT peers and directly connected peers within a 24-hour period.

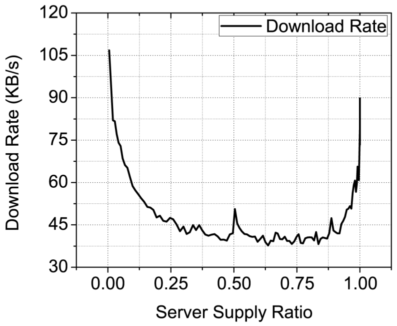


Fig. 14. The average download rate of files as a function of the server supply ratio.

server supply ratio as a ratio of the aggregate traffic supplied by servers to the aggregate download traffic of peers; 2) the *average download rate* as a ratio of the aggregate download traffic of peers to the aggregate download time of peers; and 3) the *file completion ratio* as a ratio of the aggregate download traffic of peers to the product of the file size and the number of requests from peers. Intuitively, the server supply ratio implies the level of server involvement, the average download rate represents the downloading performance of peers, and the file completion ratio reflects the satisfaction level of users.

Fig. 14 plots the average download rate of files (KB/second) as a function of the server supply ratio. We have discovered the following: 1) Most peers experienced favorable downloading rates. The average level reaches 66 KB/second and even the lowest rate is above 40 KB/second. 2) Both files that are completely supplied by servers and those that are mainly supported by P2P (with a server supply ratio below 0.1) enjoy relatively high average download rates (above 80 KB/second). 3) As we see from the valley in the curve, less popular files with server supply ratios between 0.25 to 0.8 suffer from low download rates (around 40 KB/second).

The results above reveal that the collaboration between servers and peers could potentially bring negative effects to the service quality under the current design. We believe that this is caused by the peer-side request-from-server threshold (Section 3.4) and the server-side probabilistic serving strategy (Section 3.5). In particular, for less popular files (with less peers involved), it could be difficult for a peer to achieve high P2P efficiency and download rates from partners. The reason is that even if the download rate is low, as long as the rate is above 10 KB/second (threshold), the peer can not request help from the server, which is a restriction by design. On the other hand, when the download rate is below 10 KB/second, there is no guarantee that the peer's request for help from the server be fulfilled given the probabilistic serving strategy in the design. Hence, less popular files inevitably suffer from subpar download rates in general.

Fig. 15 depicts the file completion ratio as a function of the server supply ratio. We observed that: 1) The file completion ratio increases from a bottom of 0.77 to a peak of 0.91 as the server supply ratio increases steadily; and 2) there are two remarkable jumps in the curve. One occurs when the server supply ratio reaches 1.0, while the other one occurs when the server supply ratio falls below 0.02. Since file availability can be guaranteed by servers, the major reason for a low file completion ratio is that peers suffer from low download rates and hence give up the downloading process prematurely. In this case, the decrease of average download rates shown in Fig. 14 leads to the decrease of the file completion ratio, when the server supply ratio goes down from 1.0 to 0.5. The two remarkable jumps in Fig. 15 correspond to the sharp drop and raise of average download rates when the server supply ratio is near 0 and 1. When the server supply ratio goes down from 0.5 to 0, the average download rate rises while the file completion ratio drops, which we believe is caused by the instability of peer contributions.

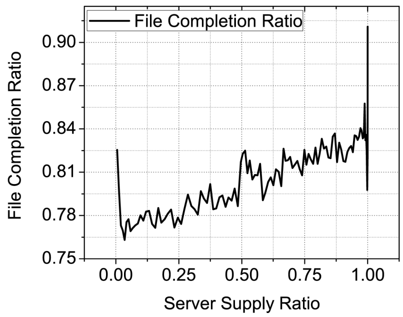


Fig. 15. The file completion ratio as a function of the server supply ratio.

5.4.2 Exploring Service Quality by File Popularity

From our existing results, we have found that the peer downloading performance is strongly correlated with the level of server involvement. Here, we further examine the service quality by taking into account both the file popularity and the time period. Fig. 16 plots the average download rates for three categories of files within a 24-hour period. We have discovered the following:

First, during the "calm" period from 2 a.m. to 8 a.m., the average download rate of popular files varies from 42 KB/second to 51 KB/second while both the average download rates of semipopular files and unpopular files stay around the level of 35 KB/second. Interestingly, at 4 a.m., there are peaks of average download rate for all three categories of files during the "calm" period. We believe these correspond to the peak of server traffic volumes at 4 a.m. as shown in Fig. 9. At 7 a.m., there is another peak of average download rate for semipopular files and unpopular files. As we can see in Fig. 12, the percentage of requests from NAT peers reaches its peak of the day and hence motivates the peers to download files directly from the servers.

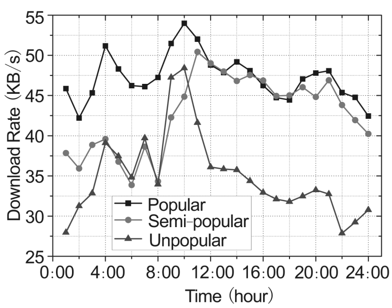


Fig. 16. The average download rates for three categories of files within a 24-hour period.

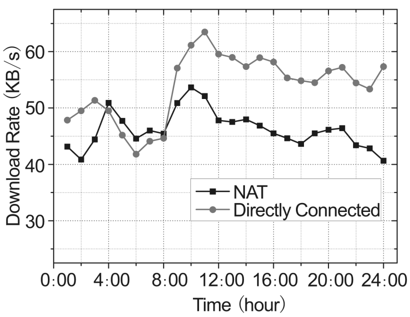


Fig. 17. The average download rates for NAT peers and directly connected peers within a 24-hour period.

Second, during the transition period from 8 a.m. to 10 a.m., the average download rates of all three categories of files increase dramatically as shown in Fig. 16. Both the average download rates of popular files and unpopular files reach their peaks at 10 a.m. while the average download rate of semipopular files reaches its peak one hour later at 11 a.m. As discussed earlier, the increases of the request-to-replica ratios of all three categories of files have motivated peers to directly download from servers, and therefore the average download rate of all three categories of files has increased at the same time.

Third, as shown in Fig. 16, during the "active" period, the average download rate of popular files and unpopular files have first decreased substantially to their previous levels during the "calm" period, and then decreased slowly as time progresses. We believe that this is caused by the server-side probabilistic serving strategy. As time progresses, the peers' requests from the server will have a lower probability to be served, and the peer download rates have declined as a result. Interestingly, though the average download rate of semipopular files decreases slowly just like the other two categories of files, the download rate stays around a relatively high level just like the popular files do, which is very different from unpopular files. The rationale is that, though semipopular files enjoy nearly the same request-to-replica ratio during the "calm" period as the "active" period as shown in Fig. 10, the number of requests is relatively high during the "active" period, in which case peer contributions are more efficient.

In summary, the server-side serving strategies work well in terms of providing peers with a high level of downloading performance. The temporary waiver of server strategy from 2 a.m. to 4 a.m. is effective in guaranteeing the peer downloading performance during the "calm" period. The server-side probabilistic serving strategy brings a negative effect in all three categories of files, but still in an acceptable range. During the "active" period, unpopular files consume most part of the server traffic, but suffer from the worst performance compared to the other two categories of files. As we have discussed above, unpopular files consist a nonnegligible portion of the entire system, and therefore should be included as one of our design objectives in our future work.

5.4.3 Exploring Service Quality by Peer Types

As a final episode, we try to examine the service quality with peer types considered (NAT versus directly connected). As shown in Fig. 17, the pattern of the average download rates of NAT peers and directly connected peers during a 24-hour period is quite similar to that of semipopular files and unpopular files as shown in Fig. 16.

During the "calm" period, NAT peers and directly connected peers enjoy similar average download rates around 45 KB/second. We believe that this is mainly affected by the server traffic volumes. As we have discussed in Section 5.3.3, the increasing percentage of NAT peers and the temporary waiver of server-side probabilistic serving strategy motivates the peers to download directly from servers. In this case, servers contribute a large portion of traffic volumes for both NAT peers and directly connected peers, and as such these two categories of peers share similar average download rates. Again, there are peaks of average download rates for both categories of peers around 3 a.m. to 4 a.m., which correspond to the peak of server traffic volumes during this period. This confirms that, during the "calm" period, servers strongly affect the service quality for both NAT peers and directly connected peers.

During the transition period from 8 a.m. to 11 a.m., the average download rates of both categories of peers have increased dramatically and reached their peaks during the day. This phenomenon is due to the increases in the number of requests and replicas, as well as in the server traffic volumes during this period. During the "active" period after 11 a.m., the average download rates of both categories of peers decrease slowly. Again, this is caused by the server-side probabilistic serving strategy.

It is interesting to observe that the daily pattern of average download rates of NAT peers and directly connected peers are quite similar to the pattern of the percentage of replicas on NAT peers as shown in Fig. 12. Both increase slowly during the "calm" period, reach their peaks of the day during the transition period, and then decrease slowly during the "active" period. Intuitively, the increase of the percentage of NAT replicas should have a negative effect on the system capacity, as well as on the service quality enjoyed by peers. However, such a negative effect has effectively motivated peers to download directly from servers, maintaining the average download rates at acceptable levels. In summary, for both NAT and directly connected peers, servers are very important in maintaining the level of service quality. The dependence on servers will, however, eventually become a bottleneck as the system scales up.

6 Conclusion

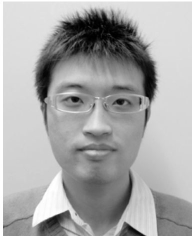
The online file hosting system has rapidly become one of the most prevailing content sharing services over the Internet due to its simplicity and versatility. Such a service is largely offered free of charge, which remains as a major attraction among the Internet users. This, however, incurs excessive bandwidth cost, and consequently results in various service restrictions. It is natural to consider leveraging bandwidth and storage contributions from peers. This paper, for the first time, describes a large scale real world peer-assisted semipersistent online file hosting system, FS2You. The fundamental challenge is to take advantage of peer bandwidth contributions and semipersistent content storage for substantial cost savings, while at the same time maintaining a high level of service availability and downloading performance at a large scale. We present the architecture and protocol design of FS2You, and demonstrate how our challenges are addressed by coupling peer assistance and server deployment in a complementary and user transparent manner. The effectiveness of the system design is verified through an extensive measurement study, which further reveals a number of interesting observations on user behavior, file characteristics, and server involvement.

ACKNOWLEDGMENTS

The authors would like to thank the editor and anonymous reviewers of the *IEEE Transactions on Parallel and Distributed Systems*, and Miss Kitty Wang from Rotman School of Management, University of Toronto, for their helpful comments and suggestions in improving the quality of the paper. The research was supported in part by grants from RGC under the contracts 615608, and 616207, by a grant from NSFC/RGC under the contract N_HKUST603/07.

REFERENCES

[1] B. Cohen, "Incentives Build Robustness in BitTorrent," *Proc. Workshop Economics of Peer-to-Peer Systems,* 2003.
[2] Rapidshare, http://www.rapidshare.com, 2010.
[3] FS2You, http://www.rayfile.com, 2010.
[4] Y. Sun, F. Liu, B. Li, B. Li, and X. Zhang, "FS2You: Peer-Assisted Semi-Persistent Online Storage at a Large Scale," *Proc. IEEE INFOCOM,* Apr. 2009.
[5] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, Analysis, and Modeling of BitTorrent-Like Systems," *Proc. ACM Internet Measurement Conference (IMC),* Oct. 2005.
[6] C. Wu, B. Li, and S. Zhao, "Multi-Channel Live P2P Streaming: Refocusing on Servers," *Proc. IEEE INFOCOM,* Apr. 2008.
[7] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, "Design and Deployment of a Hybrid CDN-P2P System for Live Video Streaming: Experiences with LiveSky," *Proc. ACM Multimedia,* Oct. 2009.
[8] Y. Huang, T. Fu, D. Chiu, J. Lui, and C. Huang, "Challenges, Design and Analysis of a Large-Scale P2P-VoD System," *Proc. ACM SIGCOMM,* Aug. 2008.
[9] C. Huang, A. Wang, J. Li, and K. Ross, "Understanding Hybrid CDN-P2P: Why Limelight Needs its Own Red Swoosh," *Proc. ACM Int'l Workshop on Network and Operating Systems Support for Digital Audio and Video (ACM NOSSDAV),* May 2008.
[10] D. Pakkala and J. Latvakoski, "Towards a Peer-to-Peer Extended Content Delivery Network," *Proc. 14th IST Mobile & Wireless Comm. Summit, Dresden,* Jun. 2005.
[11] D. Xu, S. Kulkarni, C. Rosenberg, and H. Chai, "Analysis of a CDN-P2P Hybrid Architecture for Cost-Effective Streaming Media Distribution," *Multimedia Systems,* vol. 11, no. 4, pp. 383-399, 2006.
[12] 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," *Proc. ACM Symp. Operating Systems Principles,* Oct. 2003.
[13] D. Stutzbach and R. Rejaie, "Understanding Churn in Peer-to-Peer Networks," *Proc. ACM Internet Measurement Conference (IMC),* Oct. 2006.
[14] D. Antoniades, E. Markatos, and C. Dovrolis, "One-Click Hosting Services: A File-Sharing Hideout," *Proc. ACM Internet Measurement Conference (IMC),* Nov. 2009.
[15] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," *Proc. ACM Internet Measurement Conference (IMC),* Oct. 2007.
[16] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "YouTube Traffic Characterization: A View from the Edge," *Proc. ACM Internet Measurement Conference (IMC),* Oct. 2007.
[17] B. Li, S. Xie, Y. Qu, Y. Keung, C. Lin, J. Liu, and X. Zhang, "Inside the New Coolstreaming: Principles, Measurements and Performance Implications," *Proc. IEEE INFOCOM,* Apr. 2008.
[18] B. Cheng, L. Stein, H. Jin, and Z. Zhang, "Towards Cinematic Internet Video-on-Demand," *Proc. Third ACM SIGOPS/EuroSys European Conf. Computer Systems,* Apr. 2008.
[19] L. Guo, E. Tan, S. Chen, Z. Xiao, and X. Zhang, "The Stretched Exponential Distribution of Internet Media Access Patterns," *Proc. ACM Symp. Principles of Distributed Computing (PODC),* Aug. 2008.



Fangming Liu (S'08) received the BEngr degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2005, and continued his postgraduate study and research assistant position from 2005 to 2006 at the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He is currently working toward the PhD degree at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. In 2007, he worked as a research assistant at the Department of Computer Science and Engineering, Chinese University of Hong Kong. His research interests include peer-to-peer networks, peer-assisted content distribution systems including peer-assisted online hosting systems, and peer-assisted video streaming systems. He is a student member of the IEEE.

Ye Sun received the BSci degree from the Department of Computer Science, Fudan University, China, in 2007, and the MPhil degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, in 2009. His research interests include peer-to-peer file sharing and live streaming.

Bo Li (S'89-M'92-SM'99) received the BEng degree in computer science from Tsinghua University, Beijing, and the PhD degree in the electrical and computer engineering from the University of Massachusetts at Amherst. He is a professor in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. He was with IBM Networking System, Research Triangle Park, between 1993 and 1996. He was an adjunct researcher at Microsoft Research Asia (MSRA) (1999-2006), where he spent his sabbatical leave (2003-2004). He was with Microsoft Advanced Technology Center (ATC) in the summers of 2007 and 2008. His works have resulted in more than 220 publications. He has made original contributions on Internet proxy placement, capacity provisioning in wireless networks, routing in WDM optical networks, and Internet video streaming. He is best known for a series of works on a system called Coolstreaming (Google entries over 1,000,000 in 2008 and Google scholar citations over 800), which attracted millions of download and was credited as the first large-scale Peer-to-Peer live video streaming system in the world. His recent work on the peer-assisted online hosting system, FS2You (2007-2009) (Google entries 800,000 in 2009) has also attracted millions of downloads worldwide. He received two best paper awards from IEEE. He received the Young Investigator Award from Natural Science Foundation of China (NFSC) in 2005. He has been an editor or guest editor for 17 IEEE/ACM journals and magazines, and he was involved in organizing 50 conferences. He was the Co-TPC chair for IEEE Infocom '04. He was a distinguished lecturer in IEEE Communications Society (2006-2007). He is a senior member of the IEEE.

Baochun Li (S'98-M'00-SM'05) received the BEngr degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 1995, and the MS and PhD degrees from the Department of Computer Science, University of Illinois at Urbana-Champaign, in 1997 and 2000, respectively. Since 2000, he has been with the Department of Electrical and Computer Engineering at the University of Toronto, where he is currently a professor. He holds the Bell University Laboratories Endowed chair in computer engineering since August 2005. In 2000, he was the recipient of the IEEE Communications Society Leonard G. Abraham Award in the Field of Communications Systems. In 2009, he was the recipient of the Multimedia Communications Best Paper Award from the IEEE Communications Society. His research interests include large-scale multimedia systems, peer-to-peer networks, applications of network coding, and wireless networks. He is a senior member of the IEEE, and a member of the ACM.

Xinyan Zhang (S'03) received the BS degree in computer science from Tsinghua University, Beijing, China, in 200, and the MPhil degree from the Department of Information Engineering, Chinese University of Hong Kong in 2004. He developed and implemented Coolstreaming, one of the largest Peer-to-Peer global live streaming systems at the time (2004-2005). He is with Roxbeam Corp.