# Peer-Assisted Online Storage and Distribution: Modeling and Server Strategies

Ye Sun, Fangming Liu, Bo Li
Hong Kong University of Science & Technology
{yesun, lfxad, bli}@cse.ust.hk

Baochun Li
University of Toronto
bli@eecg.toronto.edu

## ABSTRACT

Peer-assisted online storage and distribution systems have recently enjoyed large-scale deployment gaining increased popularity for multimedia content sharing in the Internet. Such systems typically deploy dedicated servers while effectively leveraging peer bandwidth in a complementary fashion, in order to guarantee adequate levels of service quality and minimize server cost. In this paper, motivated by our recent empirical study on a real-world system, FS2You, we develop a mathematical model to characterize and understand peer-assisted online storage systems serving multiple files of different popularity. Specifically, we examine and compare representative server bandwidth allocation strategies, and investigate the critical performance metrics and factors. We demonstrate that different server strategies may lead to remarkably different service qualities in terms of average downloading times, peer satisfaction levels and service quality differentiation. In particular, the current server strategy in FS2You is able to offer system-wide average downloading times comparable to the theoretical bound derived from our model.

## Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*; H.4 [**Information Systems Applications**]: Miscellaneous

## General Terms

Performance

## Keywords

Online storage, content distribution, peer-to-peer protocols, modeling studies

## 1. INTRODUCTION

Web-based online storage systems, also referred to as one-click hosting services, have rapidly become one of the most prevailing content sharing services over the Internet. Such web-based services store and distribute a variety of multimedia content to serve potentially millions of users. Via a simple URL, they allow Internet users to share files with other users, who can download them at a later time free of charge. Due to its simplicity and versatility, these online storage systems have become a favorite among users in online forums, overtaking well-known peer-to-peer (P2P) file sharing mechanisms, such as BitTorrent.

The best design philosophy for online storage systems is to take advantage of peer bandwidth contributions in a complementary fashion, even when dedicated servers are deployed to guarantee adequate levels of service quality. In our previous work, FS2You [2, 5, 8], we follow such a design philosophy to design, implement and measure a large-scale peer-assisted online storage system, which is actively used in China at the time of this writing. Supported by 350 GB worth of real-world traces from over 3.3 million users in FS2You, we have observed that around 500 GB to 1 TB of files are routinely uploaded and shared through the platform every day, among which up to 70% is accounted for by multimedia files such as videos.

From our FS2You experience, we are convinced that — very different from BitTorrent — the *number* of files being concurrently shared in peer-assisted online storage systems is very large. Such a large number of shared files are being served with highly diverse popularity, with less popular ones (with fewer peers involved) constitute more than a negligible portion of user demands. More importantly, the service quality provided by the system in terms of downloading performance is strongly correlated with file popularity, the relative proportion of popular and less popular files, and the amount of server resources allocated across these files.

These empirical observations have motivated the following important questions: (1) How do we mathematically characterize and analyze peer-assisted online storage systems with multiple files of different popularity? (2) What may be the best and most practical strategies of allocating limited server bandwidth resources across a large number of files? (3) Does the real-world server strategy used in FS2You offer acceptable performance, even if it is feasible to be implemented in practice?

In this paper, we seek to address these challenges with the following original contributions. We first develop a tractable mathematical model based on [1, 7] to characterize and understand bandwidth allocation strategies on dedicated servers, supporting multiple files in peer-assisted online storage systems. Within this theoretical framework, we are able to de-


*NOSSDAV'09,* June 3–5, 2009, Williamsburg, Virginia, USA.

rive a lower bound of the average downloading time, given a specific amount of server bandwidth in the entire system. In addition, we are able to quantitatively evaluate two representative server bandwidth allocation strategies, *request-driven* and *water-leveling*, and argue that the current server bandwidth allocation strategy adopted in FS2You is not only practical, but also offering a performance level that is comparable to the theoretical bound, which is difficult to implement in reality. Finally, we examine the effects of various influential factors, such as file popularity, the proportion of popular vs. less popular files, and the amount of server bandwidth available. We believe that the insights obtained from our study in this paper can provide important guidelines towards the best design of server bandwidth allocation strategies in practical scenarios.

## 2. RELATED WORK

There has been a number of analytical studies on BitTorrent-like P2P file sharing systems in the literature. Yang and Veciana [10] use a branching process model to study the service capacity of BitTorrent-like file sharing system in the transient regime, as well as a Markovian model for the steady-state analysis. To overcome the computation problem in [10], Qiu and Srikant [7] develop a deterministic fluid model to obtain simple expressions for the average file download time in the steady-state. Extending this work to include server provisioning, Das *et al.* [1] have discussed the motivation and effects of server participation in BitTorrent-like P2P file sharing systems. However, their service models are only restricted to the case of single file. A portion of our model is inspired by the models in [1, 7], but we further generalize them to a unified framework that takes into account both multiple files of different popularity and server bandwidth allocation across different files. While recognizing the significance of other modeling works on pure P2P environments for file sharing [3, 4, 6], our work is different from them in that we take a particular focus on the server involvement in peer-assisted online storage and distribution systems serving multiple files. More recently, based on real-world traces, Wu *et al.* [9] have proposed an online server capacity provisioning algorithm for multi-channel live P2P streaming systems. In contrast, our work is customized for peer-assisted online storage and distribution, with its own application features and performance concerns.

## 3. SYSTEM MODEL

In this section, we present our mathematical model based on [1, 7] for peer-assisted online storage systems. Different from the single-file service model in [1, 7], our model takes into account both multiple files of different popularity and server bandwidth provisioning, in order to capture the essential aspects of practical peer-assisted online storage systems such as FS2You.

### 3.1 Basic Model with Critical Factors

We consider a peer-assisted online storage system serving multiple files. There are a total of $M$ concurrent files to be served, represented as a set $F = \{1, 2, \ldots, M\}$. For any file $i \in F$, relevant notations and assumptions in our system model are summarized as follows:

$x_i(t)$: The number of peers who are downloading file $i$ in the system at time $t$. $\overline{x_i}$ is the equilibrium value of $x_i(t)$.

$y_i(t)$: The number of peers who have finished downloading file $i$ but have not yet left the system at time $t$. $\overline{y_i}$ is the equilibrium value of $y_i(t)$.

$S_i$: The server bandwidth assigned to file $i$. $S = \sum S_i$ is the total amount of server bandwidth provisioned by the service provider.

$\lambda_i$: The arrival rate of new peers in file $i$. We assume that peers arrive according to a Poisson process.

$f_i$: The size of file $i$.

$\mu$: The uploading bandwidth of a given peer. We assume that all peers have the same uploading bandwidth.

$c$: The downloading bandwidth of a given peer. We assume that all peers have the same downloading bandwidth and $c \geq \mu$.

$\theta_i$: The rate at which peers abort the download of file $i$.

$\gamma_i$: The rate at which peers who have finished downloading file $i$ leave the system.

$\eta_i$: The file sharing effectiveness of file $i$. Qiu *et al.* [7] defines $\eta$ as the fraction of the upload capacity of peers that is being utilized, with values in $[0, 1]$.

Based on [1], when the system is in steady state, the number of peers who are downloading file $i$ can be expressed as follows:

When the downloading bandwidth is the constraint, *i.e.*, if $c\overline{x_i} \leq \mu(\eta_i\overline{x_i} + \overline{y_i}) + S_i$, we have

$$\begin{aligned} \overline{x_i} &= \frac{\lambda_i}{\theta_i + \frac{c}{f_i}} \\ \overline{y_i} &= \frac{\lambda_i}{\gamma_i + \frac{\gamma_i\theta_i f_i}{c}}. \end{aligned} \tag{1}$$

When the uploading bandwidth is the constraint, *i.e.*, if $c\overline{x_i} \geq \mu(\eta_i\overline{x_i} + \overline{y_i}) + S_i$, we have

$$\begin{aligned} \overline{x_i} &= \frac{\lambda_i}{\nu_i(1 + \frac{\theta_i}{\nu_i})} - \frac{S_i}{\mu\eta_i(1 + \frac{\theta_i}{\nu_i})} \\ \overline{y_i} &= \frac{\lambda_i}{\gamma_i(1 + \frac{\theta_i}{\nu_i})} - \frac{S_i\theta_i}{f_i\gamma_i\eta_i(1 + \frac{\theta_i}{\nu_i})}, \end{aligned} \tag{2}$$

where $\frac{1}{\nu_i} = \frac{1}{\eta_i}(\frac{f_i}{\mu} - \frac{1}{\gamma_i})$.

In accordance with most of the recent Internet access technologies and measurement studies on existing P2P systems [4, 8], we assume that $c \geq \mu$ and peers will stay in the system only for a short random period of time after completing the download. Hence, the uploading bandwidth of peers in the system is most likely the constraint and we shall focus on the case of Equation (2). Furthermore, to guarantee the corresponding condition $c\overline{x_i} \geq \mu(\eta_i\overline{x_i} + \overline{y_i}) + S_i$, the amount of server bandwidth provisioned to file $i$ by the service provider shall satisfy:

$$S_i \leq \lambda_i \left( \frac{\frac{c-\mu\eta_i}{\nu_i} - \frac{\mu}{\gamma_i}}{1 + \frac{\theta_i}{\nu_i} + \frac{c-\mu\eta_i}{\mu\eta_i} + \frac{\theta_i}{\gamma_i\mu\eta_i}} \right). \tag{3}$$

To calculate the average downloading time $T_i$ for peers downloading file $i$ in steady state, we can use Little's Law as:

$$\frac{\lambda_i - \theta_i\overline{x_i}}{\lambda_i}\overline{x_i} = (\lambda_i - \theta_i\overline{x_i})T_i. \tag{4}$$

Using Equation (2), we can obtain the average downloading time of file $i$ as:

$$T_i = \frac{1}{\nu_i(1 + \frac{\theta_i}{\nu_i})}(1 - \frac{S_i\nu_i}{\lambda_i\mu\eta_i}). \tag{5}$$

Similarly, to calculate the system-wide average downloading time in steady state, we can use Little's Law again with the peer arrival rate of the entire system as $\lambda = \sum \lambda_i$ and the total number of peers in steady state as $\overline{x} = \sum \overline{x_i}$. Through Equation (2), we can obtain the system-wide average downloading time $T$ as:

$$T = \frac{\sum \overline{x_i}}{\sum \lambda_i} = \frac{1}{\sum \lambda_i} \left( \sum \frac{\lambda_i}{\nu_i(1+\frac{\theta_i}{\nu_i})} - \sum \frac{S_i}{\mu\eta_i(1+\frac{\theta_i}{\nu_i})} \right). \quad (6)$$

Observed from our recent measurement study on FS2You [8], the files in such systems can be broadly classified into two types, *popular files* and *unpopular or less popular files*. This is a coarse categorization, which is primarily determined by users' collective interests in a corresponding file; yet this categorization becomes very useful and evident in understanding the roles of servers. In a nutshell, unpopular files are mainly served by servers, while peer-assistance is pervasive for popular files. By taking this into account, we can refine the model by considering two types: *type-1* representing less popular files and *type-2* corresponding to popular files. We further assume that files of the same type share similar characteristics such as peer arrival rate and file size. This allows us to focus on server strategies, the impact from critical factors of relevance, and also helps to make the analytical model more tractable. Notations of the two types of files are subscripted by $t1$ and $t2$, respectively. For example, the numbers of the two types of files are $M_{t1}$ and $M_{t2}$, respectively. Then, Equation (6) can be rewritten as:

$$T = \frac{1}{\sum_{i=1}^{2} M_{ti}\lambda_{ti}} \left( \sum_{i=1}^{2} \frac{M_{ti}\lambda_{ti}}{\nu_{ti}(1+\frac{\theta_{ti}}{\nu_{ti}})} - \sum_{i=1}^{2} \frac{M_{ti}S_{ti}}{\mu\eta_{ti}(1+\frac{\theta_{ti}}{\nu_{ti}})} \right), \quad (7)$$

where $\lambda_{t1} \leq \lambda_{t2}$ and $\eta_{t1} \leq \eta_{t2}$ (Empirically, popular files with more peers involved usually enjoy higher file sharing effectiveness [8]). Furthermore, we assume that no peers would abort the download (*i.e.*, $\theta = 0$), and no peers would stay in the system after finishing downloading the file (*i.e.*, $\gamma \to \infty$), as a conservative (pessimistic) approximation from the service provider's perspective. Then, Equation (7) can be simplified to:

$$T = \frac{1}{M_{t1}\lambda_{t1} + M_{t2}\lambda_{t2}} \left( \frac{M_{t1}f_{t1}\lambda_{t1}\eta_{t2} + M_{t2}f_{t2}\lambda_{t2}\eta_{t1}}{\mu\eta_{t1}\eta_{t2}} - \frac{M_{t1}S_{t1}\eta_{t2} + M_{t2}S_{t2}\eta_{t1}}{\mu\eta_{t1}\eta_{t2}} \right). \quad (8)$$

Meanwhile, the corresponding condition in Equation (3) can be simplified to:

$$S_{ti} \leq (1 - \frac{\mu\eta_{ti}}{c})\lambda_{ti}f_{ti} \qquad \text{for} \quad i \in \{1,2\}, \quad (9)$$

where $S_{ti}$ denotes the server bandwidth allocated to a file of type $i$. Thus, the maximum amount of server bandwidth that can be assigned to a file of type-$i$ is $S_{maxi} = (1 - \mu\eta_{ti}/c)\lambda_{ti}f_{ti}$.

Besides, given a specific total amount of server bandwidth $S$, it is obvious that $M_{t1}S_{t1} + M_{t2}S_{t2} \leq S$. Since $S_{ti} \geq 0$, we have $M_{ti}S_{ti} \leq S$, or $S_{ti} \leq S/M_{ti}$. Finally, we have

$$S_{ti} \leq \min(S_{maxi}, \frac{S}{M_{ti}}) \qquad \text{for} \quad i \in \{1,2\}. \quad (10)$$

## 3.2 Server Bandwidth Allocation Strategies

In this section, we first derive the lower bound of the system-wide average downloading time based on the above model. Then we proceed to quantify and discuss two representative server bandwidth allocation strategies.

### 3.2.1 Lower Bound of the Average Downloading Time

To find the lower bound of the system-wide average downloading time, we first rewrite Equation (8) as:

$$T = \frac{1}{M_{t1}\lambda_{t1} + M_{t2}\lambda_{t2}} \left( \frac{M_{t1}f_{t1}\lambda_{t1}\eta_{t2} + M_{t2}f_{t2}\lambda_{t2}\eta_{t1}}{\mu\eta_{t1}\eta_{t2}} - \frac{S_{t1}(M_{t1}\eta_{t2} - M_{t2}\eta_{t1}) + SM_{t2}\eta_{t1}}{\mu\eta_{t1}\eta_{t2}} \right). \quad (11)$$

Based on our previous assumption that type-1 files are less popular while type-2 files are popular, we have $\lambda_{t1} \leq \lambda_{t2}$ and $\eta_{t1} \leq \eta_{t2}$. Furthermore, it is evident in our measurement study [8] that the proportion of less popular files is more than that of popular files in the system, *i.e.*, $M_{t1} \geq M_{t2}$. Hence, Equation (11) implies that: (1) Consistent with the result in [1], server involvement can help decrease the average downloading time compared to a pure P2P system. (2) Given a specific total amount of server bandwidth $S$, different server bandwidth allocations across the files will indeed lead to different average downloading times. The challenge, however, is how to design a near-optimal allocation strategy, that is simple enough to be implemented in practical systems. (3) To achieve the lower bound of the system-wide average downloading time, we shall first assign the server bandwidth to type-1 files until $S_{t1}$ reaches its maximum value; then, the residual server bandwidth can be assigned to type-2 files. That is,

$$\begin{aligned} S_{t1} &= \min(S_{max1}, \frac{S}{M_{t1}}) \\ S_{t2} &= \min(S_{max2}, \max(\frac{S - M_{t1}S_{max1}}{M_{t2}}, 0)). \end{aligned} \quad (12)$$

This strategy to achieve the lower bound of the system-wide average downloading time is also applicable to the general case of multiple types of files: the server should always satisfy the requests from the peers who are downloading the least popular files. However, such a strategy is hard to implement in reality, as it is hard for the system to establish *a priori* knowledge of the popularity of files and determine whether the server bandwidth allocated to a certain file has reached its maximum amount.

### 3.2.2 Request-Driven Strategy

With a request-driven strategy, the server serves every request from peers; essentially, the server bandwidth is equally divided among all the peers. We assume that the number of requests for a file to the server is proportional to the peer arrival rate of the file. We also assume that when the amount of server bandwidth assigned to one of the two types of files has reached its maximum value, the residual server bandwidth will be assigned to the other type of files. Then, we have

$$\begin{aligned} S_{t1} &= \min(S_{max1}, \max(\frac{S - M_{t2}S_{max2}}{M_{t1}}, \frac{\lambda_{t1}S}{\lambda})) \\ S_{t2} &= \min(S_{max2}, \max(\frac{S - M_{t1}S_{max1}}{M_{t2}}, \frac{\lambda_{t2}S}{\lambda})), \end{aligned} \quad (13)$$

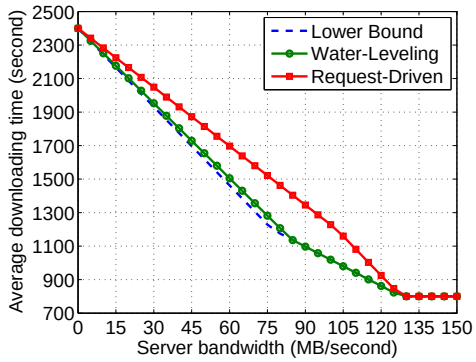
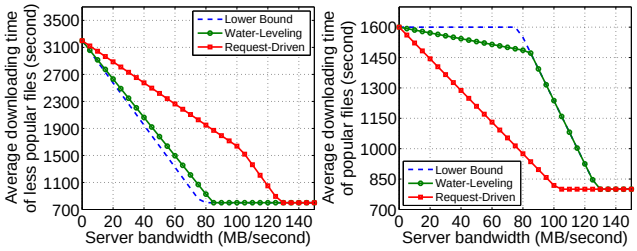



**Figure 1: Comparison of average downloading time under different server bandwidth allocation strategies, when the amount of server bandwidth varies.**




(a) (b)

**Figure 2: Comparison of average downloading time of different types of files under different server bandwidth allocation strategies, when the amount of server bandwidth varies: (a) for less popular files; (b) for popular files.**

where $\lambda = M_{t1}\lambda_{t1} + M_{t2}\lambda_{t2}$. This strategy is easy to implement in reality, as the server simply serves all the requests from peers without considering any file characteristics.

### 3.2.3 *Water-Leveling Strategy*

While the request-driven strategy equalizes the server bandwidth across all the peers, the water-leveling strategy equalizes the server bandwidth across all the files by taking file popularity into consideration. Specifically, the server serves the requests from peers according to a certain probability which is inversely proportional to the peer arrival rate of the file. If we assume that the number of requests for a file to the server is proportional to the peer arrival rate of the file, the server will serve the same number of requests for different files and therefore the server bandwidth is equally allocated across all the files. Thus, we have

$$\begin{aligned} S_{t1} &= \min(S_{max1}, \max(\frac{S - M_{t2}S_{max2}}{M_{t1}}, \frac{S}{M})) \\ S_{t2} &= \min(S_{max2}, \max(\frac{S - M_{t1}S_{max1}}{M_{t2}}, \frac{S}{M})), \quad (14) \end{aligned}$$

where $M = M_{t1}+M_{t2}$. This strategy is realized in FS2You [8], in which the server maintains a *file popularity index* for each file by recording the number of references for the file periodically, and then serves the requests from peers with a probability which is inversely proportional to the corresponding index.

## 4. NUMERICAL RESULTS AND INSIGHTS

In this section, we carry out a series of numerical analysis to compare and discuss different server bandwidth allocation strategies from different perspectives, and examine the effects of various critical factors.

### 4.1 Comparison of Server Bandwidth Allocation Strategies

We first compare the downloading performance of the system under different server bandwidth allocation strategies. Specifically, we use the following settings: for less popular files (type-1), we choose $\lambda_{t1} = 0.1$, $\eta_{t1} = 0.5$. For popular files (type-2), we choose $\lambda_{t2} = 1$, $\eta_{t2} = 1$. As mentioned earlier, this is in accordance with our measurement study [8] that less popular files have relatively lower peer arrival rates and file sharing effectiveness. Furthermore, it is also evident that the proportion of less popular files is more than that of popular files in the system; hence, we choose $M_{t1} = 10$ and $M_{t2} = 1$. Since the file size is not a major concern here, we assume that $f_{t1} = f_{t2} = 100$ MB to make the downloading performance of the two types of files comparable. Other parameters are set as $\mu = 512$ Kbps and $c = 1024$ Kbps.

Fig. 1 plots the system-wide average downloading time under request-driven and water-leveling strategies, together with the theoretical lower bound derived in Section 3.2, as a function of the amount of server bandwidth. Specifically, $S$ increases from 0 to 150 MB/second, which is thousands of times of the upload capacity of normal peers. We observe the following:

*First*, the average downloading time decreases as the server bandwidth increases under different server strategies, until the average downloading rate of peers reaches the downloading bandwidth constraint of $c = 1024$ Kbps.

*Second*, the performance of the request-driven strategy is always worse than that of the water-leveling strategy. The performance of the water-leveling strategy is very close to the lower bound and can even reach it when there is sufficient server bandwidth in the system, *e.g.*, when $S > 90$ MB/second. The rationale is that with the request-driven strategy, server bandwidth is equally divided across all the peers. Popular files with more peers involved hence occupy more server bandwidth. In contrast, with the water-leveling strategy, server bandwidth is equally divided across all the files. Since fewer peers are involved in less popular files, each of these peers can obtain more server bandwidth. This argument can be demonstrated by Fig. 2, which depicts the average downloading time of both types of files. With the request-driven strategy, as the server bandwidth increases, the average downloading time of both popular and less popular files decreases quickly, with the former decreasing faster. With the water-leveling strategy, as the server bandwidth increases, the average downloading time of less popular files decreases more rapidly, while the average downloading time of popular files decreases more slowly. According to the strategy to achieve the lower bound, the more server bandwidth allocated to less popular files, the better the system-wide performance is. These explain why the performance of the water-leveling strategy is close to the lower bound and far better than that of the request-driven strategy.

*Third*, the performance gaps between the lower bound and the two strategies first expand and then shrink as the server bandwidth increases. In other words, when the server band-

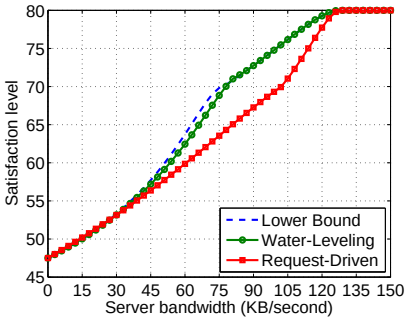



**Figure 3: Comparison of peer satisfaction level under different server bandwidth allocation strategies, when the amount of server bandwidth varies.**

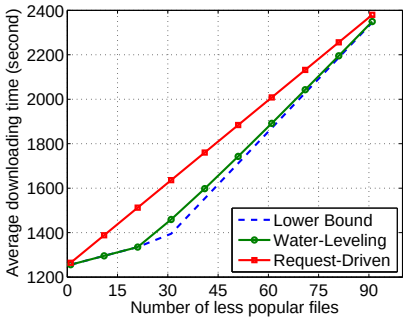



**Figure 4: Comparison of average downloading time under different server bandwidth allocation strategies, when the number of less popular files varies.**

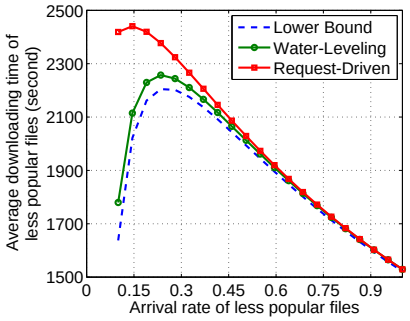



**Figure 5: Comparison of average downloading time under different server bandwidth allocation strategies, when the peer arrival rate of less popular files varies.**

width resources in the system is either constrained or sufficient, the performance gaps between different server bandwidth allocation strategies are not significant.

From the service provider's perspective, in addition to the objective of minimizing the average downloading time for the entire system, another important concern is to guarantee that as many peers as possible can enjoy a certain level of promised service quality. To this end, we introduce a metric to evaluate the peer satisfaction level as follows:

$$SL_{ti} = \alpha \min(D_{ti}, Q) + \beta \max(D_{ti} - Q, 0) \quad \text{for} \quad i \in \{1, 2\}, \tag{15}$$

where $D_{ti}$ denotes the average downloading rate of peers that are downloading type-$i$ files; $Q$ denotes the promised service quality in terms of downloading rates, which is supposed to be established by the service provider; and $\alpha$ and $\beta$ are the weights for the downloading rates below and above $Q$, respectively.

The intuition behind this metric is based on the observation that peer satisfaction should not only be directly related to the downloading rate that they obtain, but also be relevant and sensitive to the percentage gain when peers experience any downloading rate improvement. In other words, the gain in peer satisfaction level is more evident when the downloading rate jumps from 10 Kbps to 20 Kbps, as compared to when that is changed from 90 Kbps to 100 Kbps. Hence, we use $Q$ to differentiate the possible values of downloading rate into two regions, whose contributions to the peer satisfaction level are weighed by $\alpha$ and $\beta$, respectively (then, the above example can be formulated as $SL_{ti}(D_{ti} = 20) - SL_{ti}(D_{ti} = 10) > SL_{ti}(D_{ti} = 100) - SL_{ti}(D_{ti} = 90)$, given $Q = 80$ Kbps and $\alpha = 1, \beta = 0.5$). Then we can define the system-wide peer satisfaction level as: $SL = \sum_{i=1}^{2} SL_{ti} M_{ti} \lambda_{ti} / \sum_{i=1}^{2} M_{ti} \lambda_{ti}$.

To examine the peer satisfaction levels provided by different server bandwidth allocation strategies, we use the following settings: $Q = 1.2\mu$, since we expect the service quality promised by the service provider to be slightly higher than the downloading rate that peers can achieve without server provisioning. We choose $\alpha = 1$ and $\beta = 0.5$, which is sufficient to differentiate the weights of downloading rate at different levels. From Fig. 3, we observe that when the server bandwidth is constrained, the request-driven strategy can provide a better satisfaction level. However, the gaps between different strategies are very small. When the server bandwidth is not so constrained (*e.g.*, above 50 MB/second), the performance of the water-leveling strategy outperforms that of the request-driven strategy, and the gap between them is profound (*e.g.*, when $S = 84$ MB/second, the peer satisfaction level of the water-leveling strategy outperforms that of the request-driven strategy by 16%).

## 4.2 Effects from Critical Factors

### 4.2.1 *Proportion of Less Popular Files*

It is evident in [8] that the relative proportion of popular and less popular files would affect the performance of the system. Here we examine this effect by increasing the number of less popular files $M_{t1}$ from 1 to 91. In order to guarantee the number of peers arriving at the system unchanged, we also vary the number of popular files $M_{t2}$ from 10 to 1 at the same time. Server bandwidth $S$ is set to 250 MB/second. Other parameters are set as the same as in previous section.

From Fig. 4, we have made the following observations. (1) The downloading performance of the system decreases linearly as the proportion of less popular files increases. The rationale is that with the same peer arrival rate of the entire system, when less popular files become more and more dominant in the system, the uploading bandwidth contributions from peers would become lower, as the file sharing effectiveness of less popular files is lower than that of popular files. (2) When less popular files dominate the system, the server bandwidth is constrained and the performance gaps between different server strategies become smaller. This is consistent with the result in the previous section. Again, this phenomenon is caused by the fact that most of the server bandwidth is allocated to less popular files under both strategies, as less popular files crowd the system.

### 4.2.2 *File Popularity*

Our measurement studies [5, 8] on FS2You have shown that the downloading performance is strongly correlated with file popularity. Here we examine this effect by varying the peer arrival rate of less popular files (together with file sharing effectiveness), as shown in Fig. 5. Specifically, $\eta_{t1}$ increases linearly from 0.5 to 1 as $\lambda_{t1}$ increases from 0.1 to 1. We choose $S = 50$ MB/second, which is quite constrained and more similar to the case of FS2You. Other parameters are set as the same as in Section 4.1.

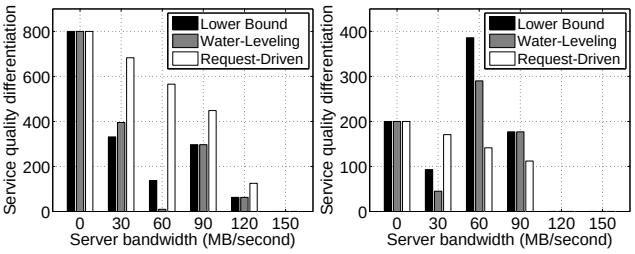



**Figure 6: Service quality differentiation under different server bandwidth allocation strategies versus the amount of server bandwidth: (a) for $\eta_{t1} = 0.5$; (b) for $\eta_{t1} = 0.8$.**

The result shown in Fig. 5 is consistent with the measurement results from FS2You. In [8], we found that both popular and less popular files experienced favorable downloading rates (above 80 KB/second), while semi-popular files suffer from low downloading rates (around 40 KB/second). In our numerical result as shown in Fig. 5, when file popularity increases, the downloading performance decreases first, and then increases again as the files become popular enough. When file popularity is low (*e.g.*, $\lambda_{t1} = 0.1$), there are fewer peers downloading the file and hence each of them can obtain relatively more server bandwidth. When file popularity is high (*e.g.*, $\lambda_{t1} > 0.6$), peers can enjoy high file sharing effectiveness and hence high average downloading rates. Semi-popular files experience the worst downloading performance in the system, as they have difficulty in both file sharing among peers and requesting help from the server. Such a U-shaped phenomenon is not observed for the request-driven strategy. The rationale is that with this strategy, the server bandwidth is equally divided across all the peers and not affected by file popularity.

## 4.3 Service Quality Differentiation

Besides the system-wide downloading performance, it is also essential to consider the differences in service quality across peers involved in files of different popularity. We use the standard deviation to quantify the service quality differentiation as: $\sigma = \sqrt{\sum_{i=1}^{2} M_{ti}\lambda_{ti}(T_{ti} - T)^2 / \sum_{i=1}^{2} M_{ti}\lambda_{ti}}$. Fig. 6 plots the above metric for $\eta_{t1} = 0.5$ and $\eta_{t1} = 0.8$, respectively. In the first case, the performance of less popular files would heavily depend on server bandwidth. In the second case, the performance of less popular files would depend less on server bandwidth.

We observe that: (1) As server bandwidth increases, the service quality differentiation keeps on falling for the request-driven strategy, as this strategy takes care of both types of files at the same time, despite the amount of server bandwidth. For the water-leveling strategy and the lower bound, the service quality differentiation first decreases then increases in scenarios when the server bandwidth is more constrained. The rationale is that both of these two strategies prefer to allocate server bandwidth to less popular files. When server bandwidth is sufficient, these two strategies will assign the residual server bandwidth to popular files and hence the service quality differentiation decreases again. (2) When less popular files have a low file sharing effectiveness (in Fig. 6(a)), the service quality differentiation of the request-driven strategy is always the worst. However, a different result is observed when less popular files have a high file sharing effectiveness (in Fig. 6(b)). In this case, when the server bandwidth is not so constrained, the request-driven strategy outperforms the other two strategies.

# 5. CONCLUSIONS

In this paper, we develop a mathematical model to analyze peer-assisted online storage and distribution systems serving multiple files of different popularity, and compare representative server bandwidth allocation strategies with regards to critical performance metrics and factors. We demonstrate that different server strategies could result in remarkably different service qualities in terms of the average downloading time, peer satisfaction level and service quality differentiation. In particular, we demonstrate that the current server strategy in a real-world system, FS2You, could offer the system-wide average downloading time comparable to the theoretical bound derived from our model; however, it may not perform well with respect to the other metrics under certain scenarios. In addition, our results have also led to a more in-depth understanding of the impact from critical factors. These observations provide valuable insights towards the design of more elaborate server strategies to improve the quality of service to end users.

# 6. ACKNOWLEDGMENTS

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

# 7. REFERENCES

[1] S. Das, S. Tewari, and L. Kleinrock. The Case for Servers in a Peer-to-Peer World. In *Proc. of IEEE ICC*, June 2006.

[2] FS2You. http://www.rayfile.com.

[3] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley. Modeling Peer-Peer File Sharing Systems. In *Proc. of IEEE INFOCOM*, March 2003.

[4] 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)*, October 2005.

[5] F. Liu, Y. Sun, B. Li, and X. Zhang. Understanding the Roles of Servers in Large-scale Peer-Assisted Online Storage Systems. In *Proc. of IEEE ICC*, June 2009.

[6] L. Massoulie and M. Vojnovic. Coupon Replication Systems. In *Proc. of ACM SIGMETRICS*, June 2005.

[7] D. Qiu and R. Srikant. Modeling and Performance Analysis of BitTorrent-like Peer-to-Peer Networks. In *Proc. of ACM SIGCOMM*, August 2004.

[8] Y. Sun, F. Liu, B. Li, B. Li, and X. Zhang. FS2You: Peer-Assisted Semi-Persistent Online Storage at a Large Scale. In *Proc. of IEEE INFOCOM*, April 2009.

[9] C. Wu, B. Li, and S. Zhao. Multi-Channel Live P2P Streaming: Refocusing on Servers. In *Proc. of IEEE INFOCOM*, April 2008.

[10] X. Yang and G. de Veciana. Service Capacity of Peer to Peer Networks. In *Proc. of IEEE INFOCOM*, March 2004.