

Free-Riding on BitTorrent-Like Peer-to-Peer File Sharing Systems: Modeling Analysis and Improvement

Minglu Li, Jiadi Yu, and Jie Wu, *Senior Member, IEEE*

Abstract—BitTorrent has emerged as a very popular peer-to-peer file sharing system, which uses an embedded set of incentive mechanisms to encourage contribution and prevent free-riding. However, BitTorrent's ability to prevent free-riding needs further study. In this paper, we present a fluid model with two different classes of peers to capture the effect of free-riding on BitTorrent-like systems. With the model, we find that BitTorrent's incentive mechanism is successful in preventing free-riding in a system without seeds but may not succeed in producing a disincentive for free-riding in a system with a high number of seeds. The reason for this is that BitTorrent does not employ any effective mechanisms for seeds to effectively guard against free-riding. Therefore, we propose a seed bandwidth allocation strategy for the BitTorrent system to reduce the effect of seeds on free-riding. Finally, simulation results are given that validate what we have found in our analysis and demonstrate the effectiveness of the proposed strategy.

Index Terms—Bandwidth allocation strategy, BitTorrent, free-riding, incentive mechanism, modeling.

1 INTRODUCTION

PEER-TO-PEER (P2P) applications have shown their popularity on the Internet for file sharing. The P2P file sharing application allows users to distribute and obtain a file to be shared cooperatively. However, most P2P collaborative systems that rely on voluntary contributions from individual participants potentially face the problem of free-riding. Free-riding behavior has the negative effect of using up the service resources of a system while contributing nothing to the system. Empirical studies [1], [2], [3] have shown that most P2P systems consequently suffer from free-riding.

Cooperation is essential to a P2P file sharing system. However, it is difficult to promote cooperation among all individual participants without an effective incentive mechanism. BitTorrent [4] is a P2P file-distribution tool that has incentive mechanisms [5] to reduce free-riding and increase user cooperation. Each peer can maximize its benefit within the constraints of the incentive mechanism. The BitTorrent system is extremely popular and is accountable for 35 percent of all of the traffic on the Internet [6].

In a BitTorrent system, a file to be shared is divided into multiple small pieces, and peers can serve other peers as soon as they have downloaded one piece of the file. In the BitTorrent system, there are two types of peers: *seeds* and *downloaders*. Seeds are peers who have all pieces of the file, whereas downloaders are peers who simultaneously download and upload pieces of the file with others. BitTorrent

employs the *tit-for-tat* peer selection strategy to prevent free-riding and promote fairness, where each peer uploads to a set of peers from which it has the highest downloading rates. In addition to the tit-for-tat strategy, BitTorrent also incorporates an *optimistic unchoking* process to probe a new connection, where each peer randomly chooses a requesting peer to upload.

A distinguishing feature of BitTorrent is its policies for cooperation and preventing free-riding. However, the effectiveness of these policies in reducing free-riding and unfairness has not yet been carefully examined under practical conditions. Some studies, [7], [8], indicated that BitTorrent mechanisms cannot prevent free-riding and unfairness. For example, Bharambe et al. [7] indicated that some peers uploaded 6.26 times as many pieces as they downloaded in BitTorrent. Jun and Ahamad [8] showed that low-bandwidth peers complete downloads in about the same amount of time as high bandwidth peers in BitTorrent. However, they did not analyze whether there was a reduction in free-riding in BitTorrent systems. In [9], Qiu and Srikant briefly discussed the effect of optimistic unchoking on free-riding and found that optimistic unchoking can induce free-riding. However, they failed to analyze the impact on free-riding that optimistic unchoking has in the BitTorrent system.

In this paper, we study the level of free-riding on BitTorrent-like P2P file sharing systems and the effect of free-riding on the performance of the BitTorrent system through a fluid model with two different classes of peers. Our contributions in this paper can be summarized as follows:

- We develop a fluid model with two different classes of peers (*non-free-riders* and *free-riders*) to capture the effect of free-riding on a BitTorrent system. With the model, we find that although optimistic unchoking may induce free-riding, free-riders do not impose a major impact through optimistic unchoking on the performance of the BitTorrent system. BitTorrent's

• M. Li and J. Yu are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200030, P.R. China. E-mail: {li-ml, jdyu}@cs.sjtu.edu.cn.

• J. Wu is with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431. E-mail: jie@cse.fau.edu.

Manuscript received 22 Nov. 2006; revised 23 June 2007; accepted 12 Sept. 2007; published online 26 Sept. 2007.

Recommended for acceptance by C. Shahabi.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0376-1106. Digital Object Identifier no. 10.1109/TPDS.2007.70783.

incentive mechanism could prevent free-riding effectively in a system without seeds.

- Applying the fluid model with two classes of peers, we study the effect of seeds on free-riding. It is seen that BitTorrent mechanisms may fail in preventing free-riding in a system having a large number of seeds. This is because free-riders can benefit significantly from seeds, and BitTorrent does not provide an effective policy for seeds to guard against free-riding.
- A seed bandwidth allocation strategy based strictly on the uploading rate of peers in the BitTorrent system is proposed. We prove that there exists a Nash equilibrium point with this strategy, under which each peer achieves its maximum uploading bandwidth. From the results of the simulation, we find that this allocation strategy not only penalizes free-riding but also quite effectively improves the performance of contributors.

The remainder of this paper is organized as follows: In Section 2, related works on free-riding and BitTorrent file sharing systems are surveyed. Section 3 provides a brief introduction for the BitTorrent system and its preliminaries. In Section 4, we analyze the incentive mechanism of BitTorrent. In Section 5, a fluid model with two classes of peers is presented to recapitulate the effect of free-riding on a BitTorrent system. A seed bandwidth allocation strategy is proposed in Section 6. In Section 7, simulation results are presented. Finally, we conclude this paper in Section 8.

2 RELATED WORK

P2P systems, as collaborative computing systems, inevitably confront the problem of free-riding. Empirical studies [1], [3] have shown prevalent free-riding in P2P file sharing systems. Research has been conducted to study free-riding on P2P file sharing systems [10], [11], [12].

Several existing P2P systems have some mechanisms built-in to encourage information sharing. For example, KaZaA [13] considers the *participation level*, which is calculated as the ratio between a peer's recent uploads and downloads. eMule [14] establishes a credit system where credits are exchanged between two specific nodes. BitTorrent systems, however, are built with information sharing as one of the main objectives. Some studies have been performed on the measurement and modeling of BitTorrent-like networks. Many measurement studies [7], [15], [16], [17], [18] based on real-world applications and simulations for BitTorrent show that the BitTorrent system has very good properties to support a large number of downloaders.

In order to understand the performance of the P2P file sharing system and BitTorrent system, many models have been presented. Ge et al. [19] and Ramachandran and Sikdar [20] present an analytic framework to study the P2P file sharing system. In [21], Yang and de Veciana discuss a branching process for studying the transient regime of the BitTorrent system and propose a Markov chain model. Qiu and Srikant [9] present a simple fluid model based on the Markov chain model proposed in [21] to describe the dynamics of the BitTorrent system. In [22], a simple mathematical model is developed, which models the behaviors of peers differently according to the state they are in. In [23] and [24], a multiclass fluid model of BitTorrent-like networks based on [9] is proposed, somewhat similar to the model discussed here. The work in [23]

focuses on parallel downloads in the case of a symmetric access link, and that in [24] studies static resource allocation for service differentiation and bandwidth diversity, which have significant differences with our work. We propose a fluid model with two classes of peers (free-riders and non-free-riders) to study free-riding behavior on BitTorrent-like networks. Our model studies the dynamic resource allocation, where the resource assignment criteria depend completely on BitTorrent mechanisms.

Several analytical studies of BitTorrent's incentive mechanisms are presented in [7], [8], [9], and [22]. In [7], it is found that BitTorrent mechanisms cannot prevent a systematical fairness through a set of simulations. Jun and Ahamad [8] provide a game-theoretic framework to explore BitTorrent's incentive mechanism. They show that free-riders are not punished properly, and contributors are not rewarded appropriately. Qiu and Srikant [9] prove that a Nash equilibrium point exists with the tit-for-tat strategy, under which each peer will upload at its maximum uploading bandwidth. Tian et al. [22] find that the original tit-for-tat strategy cannot improve file availability, and an innovative tit-for-tat strategy is proposed.

However, the capability of BitTorrent in preventing free-riding is still not fully studied. Our work differs from the above studies in that we analyze the level of free-riding found in BitTorrent systems and the impact of free-riding on the performance of the BitTorrent system through a fluid model and determine the effect of seeds on free-riding within a BitTorrent system.

3 PRELIMINARIES

BitTorrent is a P2P application that aims to enable fast and efficient distribution and downloading of large files. The basic idea in BitTorrent is to break down a shared file into equal-sized segments (typically 256 Kbytes), which are called *pieces*. A peer can download different pieces concurrently from multiple peers while uploading various pieces to other peers.

In a BitTorrent system, the sharing file provider creates a metafile called a *.torrent* file, which contains the meta-information, for example, the piece size and IP address of the so-called *tracker*, and then puts the file on a Web server. There are three components in the system: trackers, seeds, and downloaders. The tracker is a central server, which keeps track of all peers currently in the system and collects statistics to help peers find each other to exchange the file pieces. All peers in the system, including seeds and downloaders, self-organize into a P2P network, which is known as a *torrent*.

To download a file, peers download a *.torrent* file from a Web server to access the tracker and join the system. The peer asks the tracker for a list of other peers so that it can build up its peer set. The tracker then returns a random list of peers (which typically consists of 50 peers). This peer will establish connection directly to peers in the peer set, which become its neighbors. In the peer set, each peer knows the distribution of the various pieces for each peer. All the peers in the torrent will periodically report their progress to the tracker. Each peer looks for opportunities to download pieces from and upload pieces to its neighbors in its peer set. It chooses the pieces that are rarest among its neighbors in a *local rarest first* policy in order to maximize the diversity of content in the system.

BitTorrent attempts to induce fairness and guard against free-riding through a *tit-for-tat* policy. Under the tit-for-tat

policy, each peer uploads to a fixed number of other peers (the default being four) from which it could download at the highest downloading rate for a given time. The corresponding algorithm is called the *choking algorithm*. A refusal to upload to a neighbor is called *choking*, and the connections to the chosen neighbors are *unchoked*. Every 10 seconds, a peer recalculates the download rate (a rolling 20-second average) that it is receiving from its neighbor to decide who it wants to choke and who it wants to unchoke. It then leaves the situation as is until the next 10-second period is up. However, the seeds do not play by this strategy, because they are done downloading and no longer have useful download rates to decide which peers to upload to. For a seed, it will simply choose download peers to upload, which is called *upload only*. In addition to this peer selection policy, BitTorrent also incorporates an *optimistic unchoking* policy. The optimistic unchoking policies are further detailed in Section 4.

In [9], a fluid model, which is based on the Markov chain approach in [21], was developed for BitTorrent-like file sharing systems. The model assumes that all peers are homogeneous, with all peers having the same upload and download capacity. There are two states in the system: the download state and the seed state. Qiu and Srikant [9] use a Markovian description of the system in relation to the two states to develop the fluid model. The model is presented in [9], where the expressions of the numbers of downloaders and seeds and the average download time could be obtained as functions of the parameters as the peer arrival/leave rate and the upload/download rate, etc. The model gives insight as to how the average download time and the network performance of a BitTorrent-like system is affected by different parameters. The analysis proves that BitTorrent achieves very good scalability. However, the model in [9] focused only on obtaining performance indexes for homogeneous peers.

In practical applications, BitTorrent confronts the problem of free-riding, in which free-riders occupy service resources while contributing nothing. In order to capture the effect of free-riding on a BitTorrent system, we introduce a free-riding class of peers into the fluid model in [9] that only takes into account one class of peers with equal service capacity. Our model takes into account two different classes of peers: one provides service capacity, and the other contributes nothing to the system. Furthermore, we adopt dynamic resource allocation to two different classes of peers, where the resource assignment criteria depend completely on BitTorrent's mechanisms.

4 MECHANISM ANALYSIS

BitTorrent peers utilize a *tit-for-tat* strategy to select the upload/download peers: Each peer uploads to a set of peers that provide it with the highest downloading rates. This mechanism is employed to encourage the user to upload and guard against free-riding. In [9], it has been proved by the game theory that there exists a Nash equilibrium point with the *tit-for-tat* strategy under which each peer will upload at its maximum uploading bandwidth.

BitTorrent also adopts a strategy called *optimistic unchoking*. In optimistic unchoking, each peer randomly chooses a requesting downloader to upload regardless of its downloading rate, in addition to maintaining connections with those peers selected by the choking algorithm. The purpose of optimistic unchoking is that a peer could upload to

another peer that has a better downloading rate than the ones currently downloading, and the newcomer (who has no share yet) can get bootstrapped by downloading the first piece. However, the random selection of optimistic unchoking provides an opportunity for free-riders to download the file. For example, free-riders can get a downloading rate through optimistic unchoking. We need to analyze the effect of optimistic unchoking on free-riding.

Let $G = \{p_0, p_1, \dots, p_{x_n-1}, q_0, q_1, \dots, q_{x_f-1}\}$ be a set of peers in a BitTorrent system, where x_n is the number of non-free-riders, and x_f is the number of free-riders. We assume all non-free-riders have the same uploading bandwidth, and there are no seeds in G . Let μ be the uploading bandwidth of a non-free-rider. The total uploading rate of the system can be expressed as μx_n . Let u be the number of uploading connections of a non-free-rider, one of which is an optimistic unchoking uploading connection. The downloading rate of a connection is limited to $\frac{\mu}{u}$. According to optimistic unchoking, each non-free-rider randomly selects a peer to upload regardless of its downloading rate. Consequently, the total expected downloading rate of free-riders in G is

$$\begin{aligned} E[D_f] &= \sum_{k=0}^{x_n} C_{x_n}^k \left(\frac{x_f}{x_n + x_f - u} \right)^k \left(\frac{x_n - u}{x_n + x_f - u} \right)^{x_n - k} \left(k \frac{\mu}{u} \right) \\ &= \frac{x_n x_f}{x_n + x_f - u} \cdot \frac{\mu}{u} \approx \frac{x_n x_f}{x_n + x_f} \cdot \frac{\mu}{u} \end{aligned} \quad (1)$$

when $x_n + x_f \gg u$. We can see in (1) that free-riders can still get the downloading rate of $\frac{x_n x_f}{x_n + x_f} \cdot \frac{\mu}{u}$ despite the fact that they have nothing to contribute to the system. Let ρ be the ratio of the total downloading rate of free-riders to the total uploading rate of non-free-riders. We have

$$\rho = \frac{E[D_f]}{\mu x_n} = \frac{1}{u} \cdot \frac{x_f}{x_n + x_f}, \quad (2)$$

where $\rho \in [0, 1]$. We can see in (2) that free-riders may obtain a fraction of the total downloading rate of the system.

From the above analysis, we find that current BitTorrent mechanisms fail to completely eliminate free-riding, and free-riders can get service resources provided by non-free-riders through optimistic unchoking. Motivated by this observation, we first analyze the impact of free-riding to a BitTorrent system through a fluid model with two different classes of peers.

5 MODELING AND ANALYSIS

Our model is an extension of the model in [9]. In our model, download peers are divided into two classes in a BitTorrent system: *non-free-riders* and *free-riders*. Non-free-riders can provide equal service capacity, whereas free-riders contribute nothing to the BitTorrent system. In addition, seeds also provide equal service capacity to the system. We assume that free-riders will depart from the system immediately after they have finished their download and have all the pieces of the sharing file, because they do not provide any service resources to others even if they were to stay in the system. Therefore, there are three states in the system: the non-free-rider download state, the free-rider download state, and the seed state. We can obtain a Markovian description of the system in relation to the three states.

TABLE 1
Notations and Model Parameters

$x_n(t)$	number of non-free-riders in the system at time t
$x_f(t)$	number of free-riders in the system at time t
$y(t)$	number of seeds in the system at time t
λ_n	the arrival rate of the new non-free-rider
λ_f	the arrival rate of the new free-rider
μ	the uploading bandwidth of a peer, include non-free-riders and seeds
c	the downloading bandwidth of a peer, $c \geq \mu^1$
θ	the abort rate of downloaders
γ	the departure rate of seeds
η	the effectiveness of the file sharing [9]
$\rho(t)$	the ratio of the total downloading rate of free-riders to the total uploading rate of non-free-riders in the system at time t
$\kappa(t)$	the ratio of the number of free-riders to the sum number of free-riders and non-free-riders in the system at time t

¹It is realistic that the uploading bandwidth of a host is less than its downloading bandwidth, which is consistent with the current access technologies.

5.1 Modeling

A glossary of the model notations and parameters is listed in Table 1. Fig. 1 shows a general model of three states (the non-free-rider download state, the free-rider download state, and the seed state), the rate at which users flow into and flow out of the three states, and the fraction of allocated bandwidth of users in the three states on a BitTorrent file sharing system. In our model, the arrival process of the new non-free-rider and free-rider is modeled as a Poisson process with an arrival rate of λ_n and λ_f , respectively, that is, new non-free-riders and free-riders flow into the non-free-rider download state and the free-rider download state, respectively, with the rate λ_n and λ_f . The parameter η is used to indicate the efficiency of the file sharing, and it has been proved to be close to 1 in [9]. The efficiency of the file sharing of free-riders is equal to 0. At time t , the total uploading rate of the

system is $\mu(\eta x_n(t) + y(t))$. All non-free-riders and free-riders share the total uploading bandwidth provided by the non-free-riders and seeds. $\rho(t)$ gives a non-free-rider uploading bandwidth assignment criterion for free-riders. Applying the expression of (2), we have

$$\rho(t) = \frac{1}{u} \cdot \frac{x_f(t)}{x_n(t) + x_f(t)}, \quad (3)$$

where $\rho(t) \in [0, 1]$. A seed will uniformly assign its uploading bandwidth to every downloader no matter if it is the free-rider or not. Hence, the seed uploading bandwidth assignment criterion for free-riders is

$$\kappa(t) = \frac{x_f(t)}{x_n(t) + x_f(t)}, \quad (4)$$

where $\kappa(t) \in [0, 1]$. Therefore, the total downloading rate of non-free-riders is $\mu[(1 - \rho(t))\eta x_n(t) + (1 - \kappa(t))y(t)]$, and the total downloading rate of free-riders is $\mu[\rho(t)\eta x_n(t) + \kappa(t)y(t)]$. The total downloading rate of non-free-riders and free-riders cannot exceed $c x_n(t)$ and $c x_f(t)$, respectively, so we have

$$\begin{aligned} D_n(t) &= \min\{c x_n(t), \mu(1 - \rho(t))\eta x_n(t) + \mu(1 - \kappa(t))y(t)\}, \\ D_f(t) &= \min\{c x_f(t), \mu\rho(t)\eta x_n(t) + \mu\kappa(t)y(t)\}, \end{aligned} \quad (5)$$

where $D_n(t)$ and $D_f(t)$ denote the total downloading rate of non-free-riders and free-riders, respectively, at time t , that is, the rate at which non-free-riders and free-riders flow out of the non-free-rider download state and the free-rider download state, respectively, after they have finished their download. $\theta x_n(t)$ and $\theta x_f(t)$ are the rate at which non-free-riders and free-riders depart from the non-free-rider

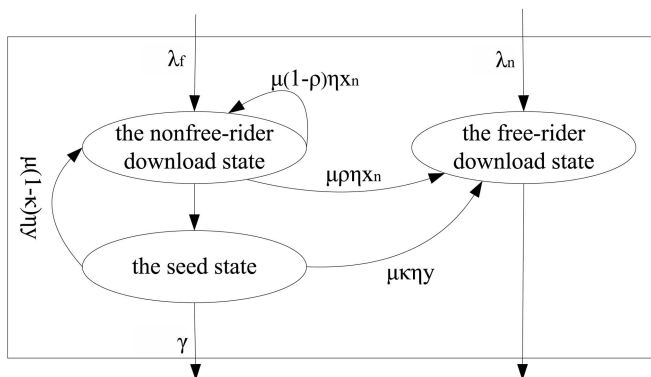


Fig. 1. General model of the three states on a BitTorrent file sharing system.

download state and the free-rider download state, respectively, without having downloaded the entire file. The non-free-rider will flow into the seed state with the rate $D_n(t)$ after they have downloaded the sharing file completely. Seeds leave the system according to an exponential distribution with the rate γ . Hence, the rate of change of the number of non-free-riders, free-riders, and seeds is given by the following equations:

$$\begin{aligned}\frac{dx_n(t)}{dt} &= \lambda_n - \theta x_n(t) - D_n(t), \\ \frac{dx_f(t)}{dt} &= \lambda_f - \theta x_f(t) - D_f(t), \\ \frac{dy(t)}{dt} &= D_n(t) - \gamma y(t).\end{aligned}\quad (6)$$

These (6) define a simple description of the evolution for the three states of the system dynamics.

5.2 Steady-State Performance Analysis and Discussion

To study the steady-state system performance, we assume that $\lim_{t \rightarrow \infty} x_n(t)$, $\lim_{t \rightarrow \infty} x_f(t)$ and $\lim_{t \rightarrow \infty} y(t)$ exist, that is

$$\lim_{t \rightarrow \infty} x_n(t) = \bar{x}_n, \lim_{t \rightarrow \infty} x_f(t) = \bar{x}_f, \lim_{t \rightarrow \infty} y(t) = \bar{y},$$

where \bar{x}_n , \bar{x}_f , and \bar{y} are the equilibrium values of $x_n(t)$, $x_f(t)$, and $y(t)$, respectively. Under the steady state $t \rightarrow \infty$, we have

$$\frac{dx_n(t)}{dt} = \frac{dx_f(t)}{dt} = \frac{dy(t)}{dt} = 0.$$

To simplify the model, we assume that the download peer will never abort the system ($\theta = 0$). We first examine the situation when the download peer will leave the system immediately upon completing the sharing-file download ($\gamma \rightarrow \infty$), that is, there are no seeds to provide uploading bandwidth in the system. We are interested in the worst situation, where peers are not willing to cooperate and provide more service capacity. Hence, the steady-state equations are given by

$$\begin{aligned}0 &= \lambda_n - \min\{c\bar{x}_n, \mu(1 - \bar{\rho})\eta\bar{x}_n\}, \\ 0 &= \lambda_f - \min\{c\bar{x}_f, \mu\bar{\rho}\eta\bar{x}_n\},\end{aligned}\quad (7)$$

where

$$\bar{\rho} = \frac{1}{u} \frac{\bar{x}_f}{\bar{x}_n + \bar{x}_f}, \quad (8)$$

where $\bar{\rho}$ is the equilibrium value of $\rho(t)$, and $\bar{\rho} \in [0, 1]$.

Theorem 1. When $c \geq \mu$ and $x_n, x_f \in [0, +\infty)$, we have $c\bar{x}_n \not\leq \mu(1 - \bar{\rho})\eta\bar{x}_n$, and $c\bar{x}_f \not\leq \mu\bar{\rho}\eta\bar{x}_n$.

Proof. If $c\bar{x}_n < \mu(1 - \bar{\rho})\eta\bar{x}_n$, we have $c < \mu\eta(1 - \bar{\rho})$ because \bar{x}_n is nonnegative. It is easy to see that $c < \mu\eta(1 - \bar{\rho}) < \mu$ because $0 \leq (1 - \bar{\rho}) \leq 1$ and $0 \leq \eta \leq 1$, which contradict with $c \geq \mu$.

If $c\bar{x}_f < \mu\bar{\rho}\eta\bar{x}_n$, we have $c < \mu\eta \frac{1}{u} \frac{\bar{x}_n}{\bar{x}_n + \bar{x}_f}$ because $\bar{\rho} = \frac{1}{u} \frac{\bar{x}_f}{\bar{x}_n + \bar{x}_f}$. It is easy to see that $c < \mu\eta \frac{1}{u} \frac{\bar{x}_n}{\bar{x}_n + \bar{x}_f} < \mu$ because $0 \leq \frac{\bar{x}_f}{\bar{x}_n + \bar{x}_f} \leq 1$ and $u \geq 1$, which contradict with $c \geq \mu$.

Therefore, Theorem 1 is true. \square

The implication of Theorem 1 is that the downloading rate is not a bottleneck for either non-free-riders or free-riders when the uploading bandwidth of a peer is less than its downloading bandwidth ($c \geq \mu$). In other words, there is no constraint on the downloading rate in a realistic situation. Solving (7) under the condition that $c\bar{x}_n \geq \mu(1 - \bar{\rho})\eta\bar{x}_n$ and $c\bar{x}_f \geq \mu\bar{\rho}\eta\bar{x}_n$, we obtain

$$\bar{x}_n = \frac{\lambda_n}{\mu\eta} \cdot \frac{1}{1 - \alpha}, \quad \bar{x}_f = \frac{\lambda_f}{\mu\eta} \cdot \frac{1}{\frac{1}{u} - \alpha}, \quad (9)$$

where $\alpha = \frac{\lambda_f}{\lambda_n + \lambda_f}$, and $\frac{1}{u} > \alpha$. In (9), we see that (7) has a unique solution, and there exists an equilibrium point (\bar{x}_n, \bar{x}_f) . However, if $\frac{1}{u} < \alpha$, the value of \bar{x}_f is negative, which does not exist in a realistic situation, that is, the free-rider does not have an equilibrium value, and $\lim_{t \rightarrow \infty} x_f(t)$ does not exist. Hence, we consider that $\frac{1}{u}$ is the threshold value of α , where the equilibrium value of free-riders exists.

Theorem 2. Let T_n and T_f be the average download time of non-free-riders and free-riders, respectively, and T be the average download time of the system. When there are no seeds in the system, we have the following results:

$$T_n = \frac{1}{\mu\eta} \cdot \frac{1}{1 - \alpha}, \quad T_f = \frac{1}{\mu\eta} \cdot \frac{1}{\frac{1}{u} - \alpha}, \quad T = \frac{1}{\mu\eta} \left[1 + \frac{1}{\frac{1}{u\alpha} - 1} \right]. \quad (10)$$

Proof. In [9], Little's law [25] was used to evaluate the average download time for a peer in the steady state as $\frac{\lambda - \theta\bar{x}}{\lambda} \bar{x} = (\lambda - \theta\bar{x})T$ (T is the average download time). Similarly, in our model, the average download time of non-free-riders and free-riders in the system is given respectively by $T_n = \frac{\bar{x}_n}{\lambda_n}$ and $T_f = \frac{\bar{x}_f}{\lambda_f}$. The probability that a peer who just completed its download job is a free-rider is $\bar{\rho}$, and the probability that it is a non-free-riding peer is $(1 - \bar{\rho})$. Therefore, the average download time of the system is given by $T = (1 - \bar{\rho})T_n + \bar{\rho}T_f$. Based on the expression of (8) and (9), the results of Theorem 2 can be easily derived. \square

The model coupled with an efficient method provides us with the ability to explore the performance of the system and capture the effect of free-riding on a BitTorrent system. Fig. 2a plots the average download time of non-free-riders, free-riders, and a system with varying values of α , given that the number of uploading connections of a peer u is 5. In Fig. 2a, we find that the average download time of free-riders T_f is always larger than the average download time of non-free-riders T_n , and there is a sharp increase in T_f with increasing α . T_n also increases, but it is not dramatic, and there is little change when α is not very large. In addition, when $\alpha \geq 0.2$, that is, $\frac{1}{u}$, the average download time of free-riders does not exist because some free-riders cannot finish their download job. This is because with an increasing α , there will be fewer peers to contribute service resources so that free-riders cannot get enough service resources to download the entire file. However, non-free-riders can always get enough service resources to finish its download job, except at $\alpha = 1$. It is shown that BitTorrent mechanisms are capable of guarding against free-riding effectively in a system without seeds, and

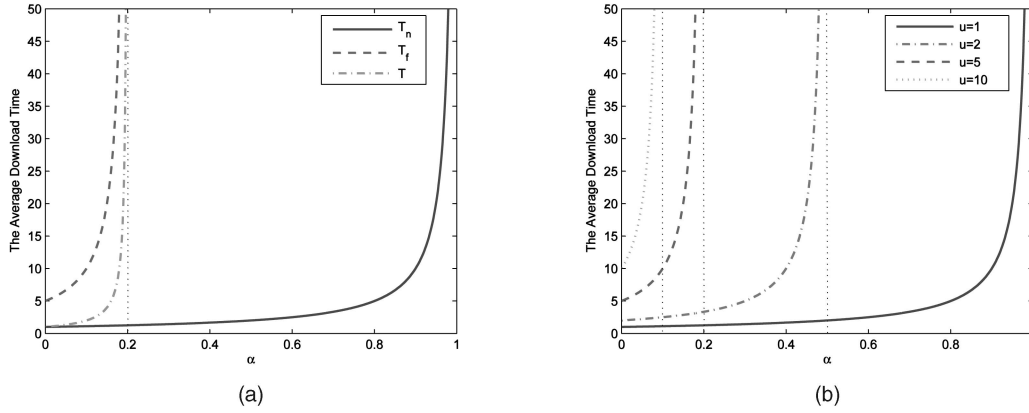


Fig. 2. (a) The average download time of non-free-riders, free-riders, and a system with varying α . (b) For different the values of u , the average download time of the non-free-rider with varying values of α .

free-riders do not impose a major impact through optimistic unchoking on the performance of non-free-riders.

Fig. 2b plots the average download time of free-riders with varying values of α when u is given a value of 1, 2, 5, or 10. From the figure, we find that as u increases, the average download time of free-riders increases as well, and the threshold value of α decreases. When $u = 1$, non-free-riders and free-riders can gain the same service resources. All free-riders can finish their download job the same as non-free-riders. However, as u increases, service resources that free-riders can gain will decrease sharply, so that it is more and more difficult for free-riders to finish their download job. For example, the value of α has to be less than 0.1 when the value of u is 10. Therefore, it is easily seen that increasing u can better guard against free-riding. However, the large number of uploading connections of a peer will lead to more time-outs and result in poor performance because multiple TCP connections have to share the same bandwidth [9]. In a BitTorrent system, the value of u is set to 5, which not only better guards against free-riding but also avoids more time-outs and poor performance, a result of multiple TCP connections.

We have previously assumed that $\gamma \rightarrow \infty$. However, in practical applications, many peers are likely to stay in the system after they have completed their download and act as a seed to serve others. Hence, free-riders can get the downloading rate from seeds to finish the download job even if BitTorrent mechanisms can prevent them completely from getting service resources from other downloaders. Based on this consideration, we will now discuss the effect of free-riding when parameters γ should be introduced. To simplify the model, we assume that each peer has a limited upload capacity, and network capacity is assumed to be unconstrained, that is, $c = \infty$ [21]. Hence, the steady-state equations are given by

$$\begin{aligned} 0 &= \lambda_n - [\mu(1 - \bar{\rho})\eta\bar{x}_n + \mu(1 - \bar{\kappa})\bar{y}], \\ 0 &= \lambda_f - (\mu\bar{\rho}\eta\bar{x}_n + \mu\bar{\kappa}\bar{y}), \\ 0 &= [\mu(1 - \bar{\rho})\eta\bar{x}_n + \mu(1 - \bar{\kappa})\bar{y}] - \gamma\bar{y}, \end{aligned} \quad (11)$$

where

$$\bar{\rho} = \frac{1}{u} \frac{\bar{x}_f}{\bar{x}_n + \bar{x}_f}, \quad \bar{\kappa} = \frac{\bar{x}_f}{\bar{x}_n + \bar{x}_f},$$

where $\bar{\rho}$ and $\bar{\kappa}$ are the equilibrium values of $\rho(t)$ and $\kappa(t)$, respectively, and $\bar{\rho}, \bar{\kappa} \in [0, 1]$. Solving (11), we obtain

$$\bar{x}_n = \frac{\lambda_n}{\mu\eta} \left(\frac{1}{1 - \alpha} - \frac{\mu}{\gamma} \right), \quad \bar{x}_f = \frac{\lambda_f}{\mu\eta} \left[\frac{1}{\frac{1}{1 - \alpha} - \frac{\mu}{\gamma}} - \left(1 - \frac{1}{\mu} \right) \right], \quad \bar{y} = \frac{\lambda_n}{\gamma} \quad (12)$$

when $\gamma > \frac{\mu}{1 - \alpha}$. We set $c = \infty$ previously. However, if the seed-leaving rate γ is smaller than $\frac{1}{1 - \alpha}\mu$, then downloading bandwidth c will determine the network performance even though c may be very large [9]. Hence, we have

$$\bar{x}_n = \frac{\lambda_n}{c}, \quad \bar{x}_f = \frac{\lambda_f}{c} \quad (13)$$

when $\gamma \leq \frac{\mu}{1 - \alpha}$. The system has an equilibrium point $(\bar{x}_n, \bar{x}_f, \bar{y})$. If $\alpha < 1 - \frac{1}{\frac{\mu}{1 - \alpha} + \gamma}$, the free-rider does not have an equilibrium value, that is, $\lim_{t \rightarrow \infty} x_f(t)$ does not exist.

Theorem 3. Let T_n and T_f be the average download time of non-free-riders and free-riders, respectively, in a BitTorrent system with seeds. We have the following results:

$$\begin{aligned} \text{when } \gamma > \frac{\mu}{1 - \alpha}, \quad T_n &= \frac{1}{\mu\eta} \left(\frac{1}{1 - \alpha} - \frac{\mu}{\gamma} \right), \\ T_f &= \frac{1}{\mu\eta} \left[\frac{1}{\frac{1}{1 - \alpha} - \frac{\mu}{\gamma}} - \left(1 - \frac{1}{\mu} \right) \right], \end{aligned} \quad (14)$$

$$\text{when } \gamma \leq \frac{\mu}{1 - \alpha}, \quad T_n = \frac{1}{c}, \quad T_f = \frac{1}{c}. \quad (15)$$

Proof. See proof of Theorem 2. \square

We know that when the departure rate of seeds decreases, the number of seeds will increase in the system, and more service resources are provided to downloaders. In Fig. 3a, we plot the ratio of the average download time of non-free-riders and free-riders with varying departure rate of seeds γ , given that the number of uploading connections of a peer u is 5. We observe that the ratio between T_n and T_f will increase when the departure rate of seeds γ decreases. When γ decreases to $\frac{1}{1 - \alpha}\mu$, free-riders and non-free-riders

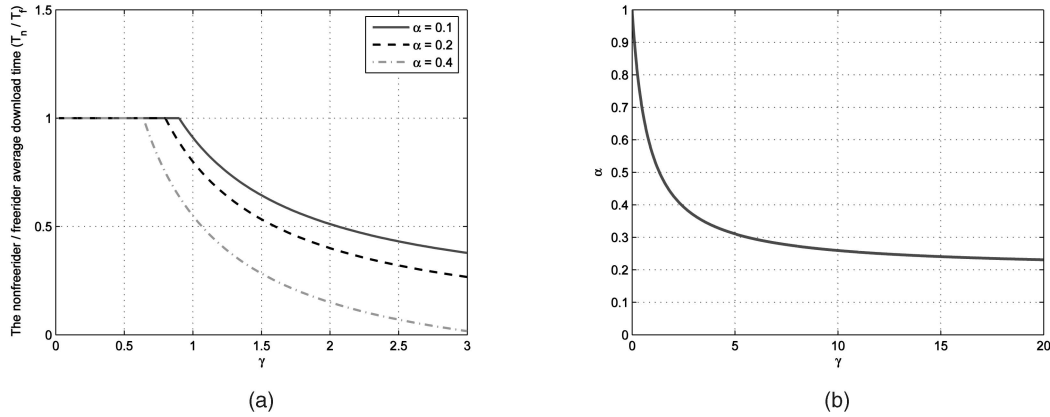


Fig. 3. (a) The ratio between the average download time of non-free-riders and free-riders as the departure rate of seeds γ varies. (b) The threshold value of α as the departure rate of seeds γ varies.

have the same average download time (the average download time is determined by downloading bandwidth c). As γ decreases, free-riders may get more service resources from seeds and download faster, whereas non-free-riders will get less service resources from seeds and download slower. It is shown that BitTorrent mechanisms may not succeed in producing a disincentive for free-riding in a system having a large number of seeds, and free-riders can get a great deal of benefit from seeds. In addition, when α increases, the ratio between T_n and T_f will decrease because the average download time of free-riders will increase just like in the system without seeds.

Fig. 3b plots the threshold value of α as the departure rate of seeds γ varies. As in a no-seed system, when α is increased to a threshold value, the free-rider cannot download the sharing file completely. In Fig. 3b, we find that the threshold value of α will increase as the departure rate of seeds γ reduces. It is shown that when there is a greater number of seeds, it is more helpful to free-riders to download the sharing file.

From the above modeling analysis, in a BitTorrent system, although optimistic unchoking can potentially result in unfairness and induce free-riding, free-riders can only obtain a few service resources under optimistic unchoking. However, the majority of service resources for free-riders is from seeds. The tit-for-tat strategy does not adapt to seeds, and there is no policy to guard against free-riding. Seeds will uniformly assign their resources to every downloader. Although seeds are volunteered to serve others whether they are free-riders or not, the potential for directly harming non-free-riders if free-riders occupy many service resources provided by seeds still exists, which is unfair to non-free-riders. In [26], the authors developed a scenario that the free-riders can completely ignore downloaders and only attempt to connect and download pieces from seeds by modifying an existing BitTorrent client. Motivated by this observation, we believe it is important for the system to establish an effective mechanism to prevent free-riders from getting more service resources from seeds.

6 PROPOSED BITTORRENT MODIFICATION

In this section, we propose a seed bandwidth allocation strategy, where seeds provide service differentiation based on the contribution of individual peers. Our strategy target is that a downloader that provides more service to the

system will be granted a higher benefit than downloaders that provide less service when some downloaders asks for a downloading file from a seed.

To provide an incentive, the seed bandwidth allocation strategy takes into account the contribution of downloaders. We define the uploading rate of downloaders as the contribution of the downloader. Decisions to allocate the bandwidth of seeds are based strictly on the current uploading rate of downloaders. When some downloaders attempt to establish connections to a seed for downloading the file, the seed retrieves the uploading rate (a rolling 20-second average uploading rate) of every requesting downloader from neighbors of downloaders through a tracker that maintains the information of neighbors of downloaders and then allocates its uploading bandwidth to requesting downloaders based on their uploading rate. Like BitTorrent's choking algorithm, each seed reallocates its uploading bandwidth every 10 seconds. To ensure the trustworthiness of feedback from neighbors of downloaders, we assume that there is a reputation mechanism in the BitTorrent system to monitor peers, such as DRBTS [27] or EigenTrust [28]. Therefore, each peer is truthful in reporting, and there is no collusion among peers. This way, no issue of false praise (over-reporting) or badmouthing (underreporting) will occur, and the neighbors will accurately report the behavior of their peers.

In the remainder of this section, we discuss how a seed implements a mechanism to distribute its uploading bandwidth among all its requesting downloaders. Note that the network capacity is assumed to be unconstrained, that is, there are no constraints on the downloading bandwidth.

6.1 Allocation Mechanism

We begin with N downloaders requesting a seed with fixed upload bandwidth. Let W be the uploading bandwidth of the seed. Downloaders that request a file download from the seed are denoted as N_1, N_2, \dots, N_N , where N is the number of requesting downloaders. The uploading bandwidth W of the seed is allocated depending on the contribution of requesting downloaders. We assume that c_i is the contribution value of the requesting downloader N_i . Then, $c = [c_1 \dots c_N]$ represents all contribution values of requesting downloaders. If $c_i \in \mathfrak{R}$, then c is a vector of N elements. Fig. 4 shows the framework of the seed bandwidth allocation.

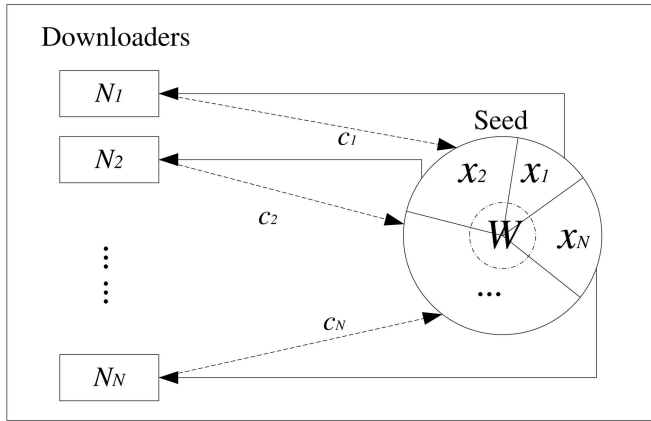


Fig. 4. Framework of the seed bandwidth allocation.

Let x_i ($x_i \geq 0$) denote the uploading bandwidth of the seed allocated to the downloader N_i , and then, $x = [x_1 \dots x_N]$ is a vector of the bandwidth allocation for all N requesting downloaders. In our design, we want our allocation to be *proportionally fair* by the contribution value c , that is, a downloader receives a downloading rate from the seed in proportion to its own uploading rate. This holds if for any other bandwidth allocation vector x' , the aggregate of the weighted proportional changes is zero or negative [29]:

$$\sum_{i=1}^N c_i \frac{x'_i - x_i}{x_i} \leq 0 \quad i = 1, 2, \dots, N,$$

where $\sum_{i=1}^N x_i = W$, and c_i denotes the weights, that is, the contribution values.

We assume that each downloader has a *utility function*, which represents the degree of satisfaction for receiving a certain allocated uploading bandwidth. Let $U_i(x_i)$ be the utility of the downloader N_i , given an uploading bandwidth allocation x_i . The utility may be a characterization of the estimated performance as a function of a given uploading bandwidth. We make the following assumption regarding U_i :

For all $i \in \{1, 2, \dots, N\}$, the utility function $U_i(x_i)$ is continuously differentiable, monotonically increasing ($U'_i(x_i) > 0$), and concave ($U''_i(x_i) \leq 0$).

Given complete knowledge, the objective for the seed bandwidth allocation can be solved as follows:

$$\begin{aligned} \max \quad & \sum_{i=1}^N U_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^N x_i = W \quad i = 1, 2, \dots, N. \end{aligned} \quad (16)$$

An optimal solution $x^* = [x_1^*, x_2^*, \dots, x_N^*]$ exists for (16), because the objective function $\sum_{i=1}^N U_i(x_i)$ is continuous, strictly increasing, and concave.

The utility function that we have chosen for downloader N_i is

$$U_i(x_i) = c_i \log(1 + x_i), \quad (17)$$

which satisfies our assumption and closely resembles the utility function of proportionally fair allocation in [29] but has $U_i(0) = 0$.

Theorem 4. *There exists the unique optimal solution $x^* = [x_1^*, x_2^*, \dots, x_N^*]$ to solve the following optimization problem:*

$$\begin{aligned} \max \quad & \sum_{i=1}^N c_i \log(1 + x_i) \\ \text{s.t.} \quad & \sum_{i=1}^N x_i = W \quad i = 1, 2, \dots, N. \end{aligned} \quad (18)$$

Proof. For all $i, i = 1, 2, \dots, N$, we have

$$\begin{aligned} \frac{\partial^2 U_i}{\partial x_i^2} &= -\frac{c_i}{(1 + x_i)^2} < 0 \quad i = 1, 2, \dots, N, \\ \frac{\partial^2 U_i}{\partial x_i \partial x_j} &= 0 \quad i, j = 1, 2, \dots, N. \end{aligned}$$

The Hessian matrix of the utility function U_i is

$$\begin{aligned} \nabla^2 U_i &= \begin{pmatrix} \frac{\partial^2 U_1}{\partial x_1^2} & \frac{\partial^2 U_1}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 U_1}{\partial x_1 \partial x_N} \\ \frac{\partial^2 U_2}{\partial x_2 \partial x_1} & \frac{\partial^2 U_2}{\partial x_2^2} & \dots & \frac{\partial^2 U_2}{\partial x_2 \partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 U_N}{\partial x_N \partial x_1} & \frac{\partial^2 U_N}{\partial x_N \partial x_2} & \dots & \frac{\partial^2 U_N}{\partial x_N^2} \end{pmatrix} \\ &= \begin{pmatrix} -\frac{c_1}{(1+x_1)^2} & 0 & \dots & 0 \\ 0 & -\frac{c_2}{(1+x_2)^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -\frac{c_N}{(1+x_N)^2} \end{pmatrix}. \end{aligned}$$

It is now easy to see that $\nabla^2 U_i$ is negative definite, and thus, U_i is strictly concave. Therefore, the optimization problem (18) allows a unique optimal solution.

We have the following Lagrangian function:

$$L(x, \lambda) = \sum_{i=1}^N c_i \log(1 + x_i) - \lambda \left(\sum_{i=1}^N x_i - W \right),$$

where λ is the Lagrangian multiplier.

$$\begin{aligned} \frac{\partial L}{\partial x_i} &= \frac{c_i}{(1 + x_i^*)} - \lambda = 0 \quad i = 1, 2, \dots, N, \\ \frac{\partial L}{\partial \lambda} &= \sum_{i=1}^N x_i^* - W = 0. \end{aligned}$$

There exists a nonnegative Lagrangian multiplier λ such that the above conditions are satisfied:

$$\lambda = \frac{c_i}{(1 + x_i^*)} \quad i = 1, 2, \dots, N.$$

When $x_i^* \geq 0, i = 1, 2, \dots, N$, it follows that

$$\frac{c_i}{(1 + x_i^*)} = \frac{c_j}{(1 + x_j^*)} \quad i, j = 1, 2, \dots, N.$$

This can be rewritten as

$$\frac{(1 + x_j^*)}{(1 + x_i^*)} = \frac{c_j}{c_i} \quad i, j = 1, 2, \dots, N.$$

To determine the bandwidth allocation strategy, we have

$$\sum_{k=1}^N \frac{(1+x_k^*)}{(1+x_i^*)} = \sum_{k=1}^N \frac{c_k}{c_i}, \quad \text{that is,} \quad \frac{(1+x_i^*)}{\sum_{k=1}^N (1+x_k^*)} = \frac{c_i}{\sum_{k=1}^N c_k},$$

and then, x_i^* can be expressed as

$$x_i^* = \frac{c_i}{\sum_{k=1}^N c_k} \left(\sum_{k=1}^N x_k^* + N \right) - 1 = \frac{c_i}{\sum_{k=1}^N c_k} (W + N) - 1. \quad (19)$$

x_i^* ($i = 1, 2, \dots, N$) is the unique optimal solution of the optimization problem (18). \square

Equation (19) provides a bandwidth allocation policy of the seed among all requesting downloaders. In (19), since $x_i^* \geq 0$, we note that c_i should not be too small and far from the average contribution value. The downloader who provides a smaller contribution value will be dropped. If $\exists x_i < 0$, $i \in \{1, 2, \dots, N\}$, the requesting downloader N_i will be dropped, and then, the seed reallocates its uploading bandwidth to other requesting downloaders except N_i . This step will be repeated until all allocated uploading bandwidths are not smaller than zero, that is, $x_i \geq 0$, $i = 1, 2, \dots, N$.

The seed bandwidth allocation policy

Instance: All requesting downloaders set $N = \{N_1, N_2, \dots, N_N\}$, the contribution value set $c = [c_1 \dots c_N]$, and the bandwidth allocation set $x = [x_1 \dots x_N]$.

1. The seed retrieves the uploading rate c of every requesting downloader N_i , $N_i \in N$ from the neighbors of downloaders.
2. The seed assigns its uploading bandwidth W to the requesting downloader N_i , $\forall N_i \in N$ according to the (19) and gets x_i , $x_i \in x$.
3. If $\exists x_i < 0$, $x_i \in x$, then $N = N - N_i$, and $x_i = 0$.
4. Repeat step 2 until $\forall x_i \geq 0$, $x_i \in x$.

Note that the requesting downloader whose contribution value is zero (that is, contributes nothing to the system) will be dropped for certain and cannot obtain any downloading rate from the seed. Thus, the contribution value of downloaders must be larger than zero (at least) in order to obtain a downloading rate from the seed, that is, $c_i > 0$.

Theorem 5. For any two requesting downloaders N_i , N_j ($i, j \in \{1, 2, \dots, N\}$), we have

$$\text{if } c_i \geq c_j, \quad \text{then } U_i(x_i) \geq U_j(x_j).$$

Proof. According to (19), if $c_i \geq c_j$, then $x_i \geq x_j$. Therefore, $(1+x_i) \geq (1+x_j) \Rightarrow \log(1+x_i) \geq \log(1+x_j) \Rightarrow c_i \log(1+x_i) \geq c_j \log(1+x_j) \Rightarrow U_i(x_i) \geq U_j(x_j)$. \square

In Theorem 5, we know that the seed bandwidth allocation policy provides a higher utility for requesting downloaders who have the higher uploading rate. Therefore, our allocation policy provides an incentive in the BitTorrent system.

6.2 Nash Equilibrium

We now consider whether all requesting downloaders are satisfied under our seed bandwidth allocation policy. In the game theory, this is determined by seeing whether there exists a Nash equilibrium $c^* = [c_1^*, c_2^*, \dots, c_N^*]$, where $c^* > 0$. At the Nash equilibrium, no single downloader wishes to deviate from its contribution value or has an incentive to change its strategy because the contribution value of each requesting downloader is the best response to the contribution value of other requesting downloaders.

We adopt the notation c_{-i} to denote the vector of all requesting downloaders other than c_i , that is, $c_{-i} = [c_1, c_2, \dots, c_{i-1}, c_{i+1}, \dots, c_N]$. Suppose that all requesting downloaders have the same physical uploading bandwidth μ (that is, maximum uploading bandwidth), and then, $c_i \in [0, \mu]$, $\forall i \in \{1, 2, \dots, N\}$. A Nash equilibrium of a game defined by (U_1, U_2, \dots, U_N) for all requesting downloaders N_i is

$$U_i(c_i^*; c_{-i}^*) \geq U_i(c_i; c_{-i}^*),$$

where $c_i \in [0, \mu]$, $i = 1, 2, \dots, N$, and $c_{-i}^* > 0$.

Because the utility function $U_i(x_i)$ is continuously differentiable, monotonically increasing, and concave, every requesting downloader's optimal response is captured in its bandwidth allocation function $x_i^*(c)$, that is, (19). Therefore, we can use $x_i^*(c)$ as a tool to evaluate the existence of a Nash equilibrium. Therefore, we have

$$x_i(c_i^*; c_{-i}^*) \geq x_i(c_i; c_{-i}^*),$$

where $c_i \in [0, \mu]$, $i = 1, 2, \dots, N$, and $c_{-i}^* > 0$.

Therefore, the game defined by (U_1, U_2, \dots, U_N) can be expressed as the following constraint optimization problem:

$$\begin{aligned} \max \quad & x_i(c_i, c_{-i}^*) = \frac{c_i}{c_i + \sum c_{-i}^*} (W + N) - 1 \\ \text{s.t.} \quad & c_i \in [0, \mu] \quad i = 1, 2, \dots, N. \end{aligned} \quad (20)$$

Theorem 6. $x_i(c)$ ($i = 1, 2, \dots, N$) is a continuous function of $c > 0$. For any $c_{-i}^* > 0$, $x_i(c)$ is strictly increasing and concave in $c_i \in [0, \mu]$.

Proof. From (19), we know that $x_i(c_i, c_{-i}^*)$ is continuously differentiable in $c_i \geq 0$. For any $c_{-i}^* > 0$, we have

$$\begin{aligned} \frac{\partial x_i(c_i, c_{-i}^*)}{\partial c_i} &= (W + N) \left[\frac{1}{c_i + \sum c_{-i}^*} - \frac{c_i}{(c_i + \sum c_{-i}^*)^2} \right] > 0, \\ \frac{\partial^2 x_i(c_i, c_{-i}^*)}{\partial c_i^2} &= (W + N) \left[\frac{2c_i}{(c_i + \sum c_{-i}^*)^3} - \frac{2}{(c_i + \sum c_{-i}^*)^2} \right] < 0. \end{aligned}$$

Thus, $x_i(c_i, c_{-i}^*)$ is strictly increasing and concave for $c_i \in [0, \mu]$, which implies that $x_i(c_i, c_{-i}^*)$ has a unique optimal solution c_i^* in $c_i \in [0, \mu]$, where $i = 1, 2, \dots, N$, that satisfies $x_i(c_i; c_{-i}^*) < x_i(c_i^*; c_{-i}^*)$ for $\forall c_i \in [0, \mu]$ and $c_i \neq c_i^*$. \square

Theorem 6 establishes the concavity and continuity of $x_i(c)$, where $i = 1, 2, \dots, N$, which guarantees the existence of a Nash equilibrium c^* for the game.

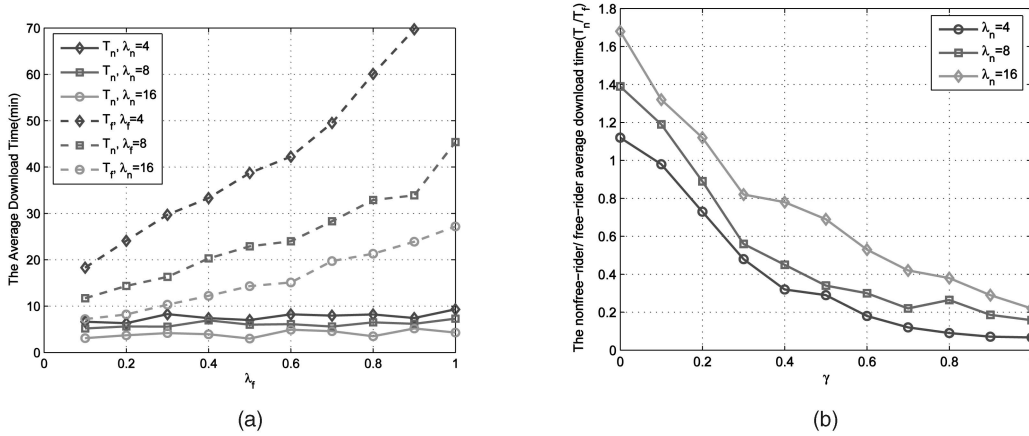


Fig. 5. (a) The average download time of non-free-riders T_n and free-riders T_f with varying arrival rates of free-riders λ_f , and $\gamma \rightarrow \infty$. (b) The ratio between the average download time of non-free-riders and free-riders T_n/T_f as the departure rate of seeds γ varies.

Theorem 7. The strategy $c_i^* = \mu$ for the requesting downloader N_i , where $i = 1, 2, \dots, N$, is a Nash equilibrium.

Proof. Let us consider the constraint optimization problem (20). We have the following Lagrangian function:

$$L_i(c_i, \lambda) = \frac{c_i}{c_i + \sum c_i^*} (W + N) - 1 - \lambda(c_i - \mu),$$

where λ is the Lagrangian multiplier. The Kuhn-Tucker condition requires that there exists a nonnegative Lagrangian multiplier to satisfy the following conditions:

$$\frac{\partial L_i}{\partial c_i} = (W + N) \left[\frac{1}{\sum_{i=1}^N c_i^*} - \frac{c_i^*}{\left(\sum_{i=1}^N c_i^*\right)^2} \right] - \lambda = 0,$$

$$0 = \lambda(c_i^* - \mu).$$

If $c_i^* < \mu$, then $\lambda = 0$, and then, $\sum_{i=1}^N c_i^* = c_i^*$. However, we have $c_{-i}^* > 0$, which is a contradiction. Therefore, an optimal solution for $c_i^* < \mu$ does not exist. If $c_i^* = \mu$, there exists a nonnegative Lagrangian multiplier λ , and a unique optimal solution exists in $c_i^* = \mu$. For any N_i , where $i = 1, 2, \dots, N$, if $c_i < \mu$ and $c_i \in [0, \mu]$, then $x_i(c_i; c_{-i}^*) < x_i(\mu; c_{-i}^*)$. Therefore, the strategy $c_i^* = \mu$ for the requesting downloader N_i , where $i = 1, 2, \dots, N$, is a Nash equilibrium. \square

To maximize the download rate from seeds, each downloader will increase its uploading bandwidth to the system until it reaches its maximum limit. At that point, it will refrain from changing its uploading bandwidth. When each requesting downloader achieves its maximum uploading bandwidth, there exists a Nash equilibrium. Therefore, our allocation policy implements an incentive for downloaders to contribute its uploading bandwidth to the system.

7 SIMULATION RESULTS

In this section, we present the results generated by performing two sets of simulations. Our purpose is to

validate our analysis and support our seed bandwidth allocation strategy, as discussed in Sections 5 and 6.

7.1 Model Validation

In this simulation, we study the results from a discrete-event simulation of a BitTorrent-like network. In the simulated network, we allow peers to dynamically join or leave the system. The arrival process of peers is a Poisson process. A peer can depart from the system after finishing their download and obtaining all pieces of the sharing file. We set the served file size as 50 Mbytes, which is divided into 200 pieces and 256 Kbyte per piece. The uploading bandwidth of non-free-riders and seeds is set as 500 Kbps, and there are no constraints on the downloading bandwidth. The number of concurrent upload transfers of each peer is 5. One initial seed is inserted into the system in order to bootstrap the system, and 1,000 downloaders will join the system.

In Fig. 5a, we plot the average download time of non-free-riders T_n and free-riders T_f with varying arrival rates of free-riders λ_f and set $\lambda_n = 4, 8, 16$, respectively. We set $\gamma \rightarrow \infty$, that is, non-free-riders will leave the system at once as soon as they have downloaded the file completely. In the figure, we can see that the average download time of free-riders T_f increases sharply, whereas the average download time of non-free-riders T_n remains nearly unchanged as the value of λ_f increases or the value of λ_n decreases, which implies that BitTorrent mechanisms are successful in penalizing free-riding, in effect by increasing the download time of free-riders, which supports our modeling results.

In Fig. 5b, we set $\lambda_n = 4, 8, 16$, respectively, and $\lambda_f = 1$. It can be observed that with the decrease of the departure rate of seeds γ , the ratio between the average download time of non-free-riders and free-riders will increase. Free-riders will download faster as the number of seeds in the system increases and may even download faster than non-free-riders with a high number of seeds, as shown in our modeling results in Fig. 3a. We believe that this is because the BitTorrent system does not provide an effective mechanism for seeds to guard against free-riding. Moreover, when λ_n increases, the ratio between T_n and T_f will increase, as our modeling analysis shows.

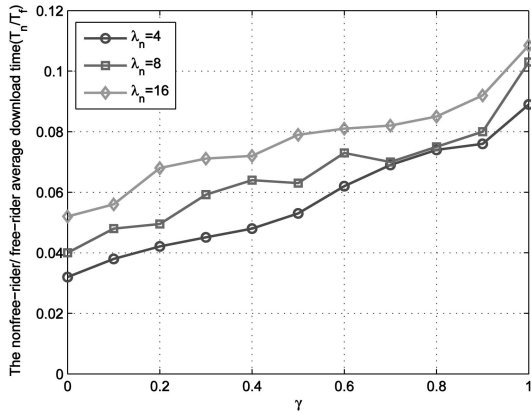


Fig. 6. The ratio between the average download time of non-free-riders and free-riders T_n/T_f as the departure rate of seeds γ varies under each seed playing the seed bandwidth allocation strategy.

7.2 Impact of the Seed Bandwidth Allocation Strategy

In this simulation, we use the same discrete-event simulator in the first simulation to study the seed bandwidth allocation strategy based strictly on the contribution of downloaders in a BitTorrent-like network. Our purpose is to evaluate the performance of the system when the seed employs the bandwidth allocation strategy and the original BitTorrent mechanisms.

We have the same setting as the first simulation. From Fig. 6, we find that when the seed bandwidth allocation strategy is employed, the ratio between the average download time of non-free-riders and free-riders will decrease as the departure rate of seeds γ decreases. Furthermore, we observe that the average download time of non-free-riders is apparently smaller than that of free-riders, that is, non-free-riders always download faster than free-riders regardless of the number of seeds.

We compare the effect of the two strategies, that is, our bandwidth allocation strategy and the original BitTorrent mechanisms, on the average download time of free-riders and non-free-riders, and the results are shown in Figs. 7a and 7b. We set $\lambda_n = 8$ and $\lambda_f = 1$. From Fig. 7a, we find that the average download time of free-riders is not shortened with the increase of the number of seeds by employing the

seed bandwidth allocation strategy, and the average download time of free-riders is increased compared with that using original BitTorrent mechanisms, which implies that the seed bandwidth allocation strategy is successful in penalizing free-riding and preventing free-riders from getting the downloading rate from seeds. In Fig. 7b, it is observed that the average download time of non-free-riders with the seed bandwidth allocation strategy is apparently shorter than that without this strategy, and non-free-riders will download faster with a high number of seeds. From the result found in the simulation, we can see that the seed bandwidth allocation strategy not only penalizes free-riding but also is helpful to contributing peers.

In the above simulations, we study the impact of the seed bandwidth allocation strategy in a setting consisting of a homogeneous collection of non-free-riders. In this simulation, we evaluate the seed bandwidth allocation strategy when the non-free-rider bandwidth is heterogeneous. The uploading bandwidth of seeds is 500 Kbps, and the upload rates of non-free-riders are distributed uniformly over [200, 500]. We compare the effect of our bandwidth allocation strategy and the original BitTorrent mechanisms on the average download time of free-riders and non-free-riders, and the results are shown in Figs. 8a and 8b. In Fig. 8, we can see that just as in a homogeneous environment, the average download time of free-riders is not shortened with an increase in the number of seeds by employing the seed bandwidth allocation strategy (Fig. 8a), whereas non-free-riders will download faster than that without this strategy (Fig. 8b), when the non-free-rider bandwidth is heterogeneous. Therefore, the seed bandwidth allocation strategy can prevent free-riding effectively and improves the performance of contributors not only in homogeneous environments but also in heterogeneous environments.

We compare the average download time of peers with various uploading bandwidths. We set that there are an equal number of peers with 100 Kbps, 200 Kbps, 500 Kbps, and 800 Kbps uploading bandwidth. The uploading bandwidth of seeds is 500 Kbps, and seeds do not leave the system ($\gamma = 0$). From Fig. 9, we find that the average download time of peers with the high uploading bandwidth is shorter than that of peers with the low uploading bandwidth. The average download time of peers with the high uploading

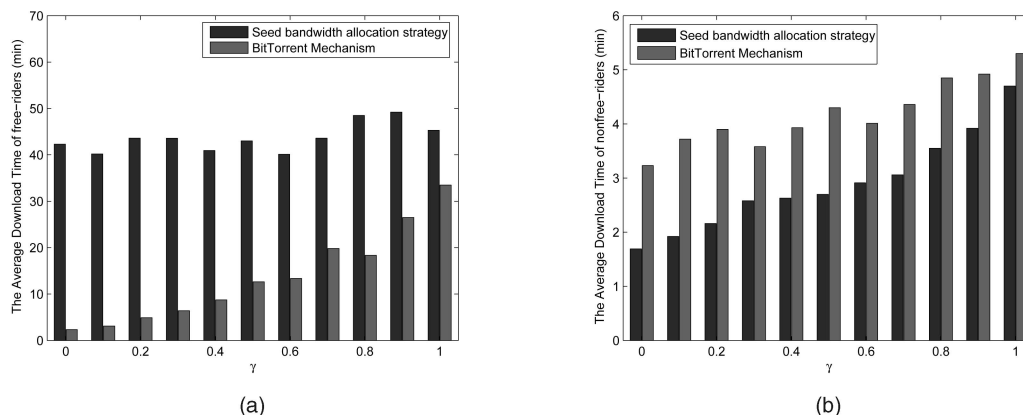


Fig. 7. The average download time of (a) free-riders and (b) non-free-riders under each seed playing the seed bandwidth allocation strategy and BitTorrent mechanisms.

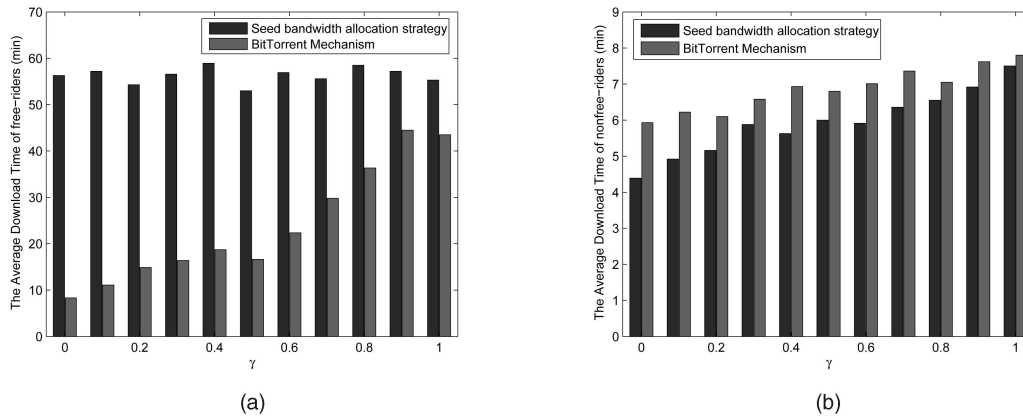


Fig. 8. In heterogeneous environment, the average download time of (a) free-riders and (b) non-free-riders under each seed playing the seed bandwidth allocation strategy and BitTorrent mechanisms.

bandwidth is shortened, whereas the average download time of peers with the low uploading bandwidth is prolonged by employing the seed bandwidth allocation strategy. From the results found in the simulation, we can see that the seed bandwidth allocation strategy implements an incentive to peers to contribute services to the system.

8 CONCLUSION

In this paper, we first investigate the choking algorithm and optimistic unchoking of the BitTorrent mechanisms. We find that BitTorrent mechanisms could not completely eliminate free-riding. Free-riders can still get a downloading rate from contributors through optimistic unchoking. To further elucidate the effect of free-riding, we developed a fluid model with two different classes of peers. We find that the effect of optimistic unchoking on free-riding does not significantly impact the performance of the BitTorrent system. BitTorrent's incentive mechanism is successful in preventing free-riding in a system without seeds. However, BitTorrent mechanisms may not succeed in producing a disincentive for free-riding in a system having a large number of seeds, because free-riders can get a great deal of benefit from seeds and BitTorrent does not have an effective mechanism for seeds to guard against free-riding.

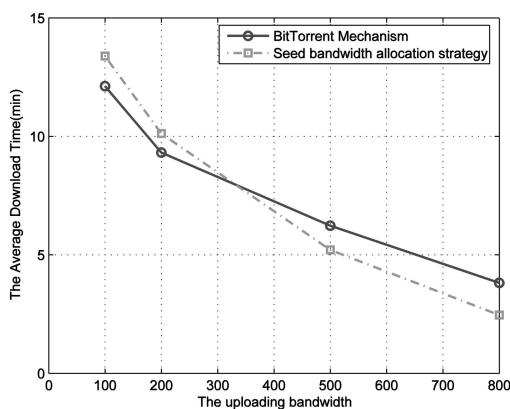


Fig. 9. The average download time of peers with various uploading bandwidths.

Thus, we present a seed bandwidth allocation strategy based strictly on the uploading rate of peers in the BitTorrent system to prevent free-riders from getting benefit from seeds. Our simulation results validate our analysis and show that the seed bandwidth allocation strategy not only guards against free-riding effectively but also quite effectively improves the performance of contributors.

In the future, we plan to extend our model to heterogeneous peers with different utility functions. We will also conduct more exhaustive simulations to confirm the robustness of the seed bandwidth allocation strategy. Further, we will validate our analysis and evaluate the proposed policy through empirical experimentation under a real environment.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grants 60628201, 60473092, 60433040, and 90612018, and by the National Basic Research Program of China under Grant 2007CB310905.

REFERENCES

- [1] E. Adar and B.A. Huberman, "Free-Riding on Gnutella," *First Monday*, vol. 5, no. 10, Oct. 2000.
- [2] D. Hughes, G. Coulson, and J. Walkerdine, "Free-Riding on Gnutella Revisited: The Bell Tolls," *IEEE Distributed Systems Online*, vol. 6, no. 6, 2005.
- [3] S. Saroiu, P.K. Gummadi, and S.D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," *Proc. Conf. Multimedia Computing and Networking (MMCN '02)*, Jan. 2002.
- [4] BitTorrent, Inc., *BitTorrent Web Site*, <http://www.bittorrent.com>, 2001-2005.
- [5] B. Cohen, "Incentives Build Robustness in BitTorrent," *Proc. ACM SIGCOMM Workshop Economics of Peer-to-Peer Systems (P2PECON)*, 2003.
- [6] A. Parker, *The True Picture of Peer-to-Peer File Sharing*, <http://www.cachelogic.com/>, 2004.
- [7] A.R. Bharambe, C. Herley, and V.N. Padmanabhan, "Analyzing and Improving BitTorrent Performance," Technical Report MSR-TR-2005-03, Microsoft Research, Feb. 2005.
- [8] S. Jun and M. Ahamad, "Incentives in BitTorrent Induce Free Riding," *Proc. ACM SIGCOMM Workshop Economics of Peer-to-Peer Systems (P2PECON '05)*, Aug. 2005.
- [9] D. Qiu and R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks," *Proc. ACM SIGCOMM '04*, Aug. 2004.

- [10] M. Yang, Z. Zhang, X. Li, and Y. Dai, "An Empirical Study of Free-Riding Behavior in the Maze P2P File-Sharing System," *Proc. Fourth Int'l Workshop Peer-to-Peer Systems (IPTPS '05)*, Feb. 2005.
- [11] R. Krishnan, M. Smith, Z. Tang, and R. Telang, "The Virtual Commons: Why Free-Riding Can Be Tolerated in Peer-to-Peer Networks," *Proc. Workshop Information Systems and Economics (WISE '03)*, Dec. 2003.
- [12] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-Riding and Whitewashing in Peer-to-Peer Systems," *Proc. ACM SIGCOMM Workshop Practice and Theory of Incentives in Networked Systems (PINS '04)*, Aug. 2004.
- [13] *KaZaA*, <http://www.kazaa.com>, 2006.
- [14] *eMule*, <http://www.emule-project.net>, 2006.
- [15] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, Analysis, and Modeling of BitTorrent-Like Systems," *Proc. Internet Measurement Conf. (IMC '05)*, Oct. 2005.
- [16] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. Hamra, and L. Garcés-Erice, "Dissecting BitTorrent: Five Months in a Torrents Lifetime," *Proc. Fifth Passive and Active Measurements Workshop (PAM '04)*, Apr. 2004.
- [17] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips, "A Measurement Study of the BitTorrent Peer-to-Peer File-Sharing System," Technical Report PDS-2004-003, Delft Univ. of Technology, The Netherlands, Apr. 2004.
- [18] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips, "The BitTorrent P2P File-Sharing System: Measurements and Analysis," *Proc. Fourth Int'l Workshop Peer-to-Peer Systems (IPTPS '05)*, Feb. 2005.
- [19] Z. Ge, D.R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling Peer-to-Peer File Sharing Systems," *Proc. IEEE INFOCOM*, 2003.
- [20] K. Ramachandran and B. Sikdar, "An Analytic Framework for Modeling Peer to Peer Networks," *Proc. IEEE INFOCOM '05*, Mar. 2005.
- [21] X. Yang and G. de Veciana, "Service Capacity of Peer to Peer Networks," *Proc. IEEE INFOCOM*, 2004.
- [22] Y. Tian, D. Wu, and K.-W. Ng, "Modeling, Analysis and Improvement for BitTorrent-Like File Sharing Networks," *Proc. IEEE INFOCOM*, 2006.
- [23] F.L. Piccolo, G. Neglia, and G. Bianchi, "The Effect of Heterogeneous Link Capacities in BitTorrent-Like File Sharing Systems," *Proc. Int'l Workshop Hot Topics in Peer-to-Peer Systems (HOT-P2P)*, 2004.
- [24] F. Clvenot-Perronnin and K.R.P. Nain, "Multiclass P2P Networks: Static Resource Allocation for Service Differentiation and Bandwidth Diversity," *Proc. 24th Int'l Symp. Computer Performance, Modeling, Measurements and Evaluation (Performance)*, 2005.
- [25] D. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 1987.
- [26] N. Liogkas, R. Nelson, E. Kohler, and L. Zhang, "Exploiting BitTorrent for Fun (but Not Profit)," *Proc. Fifth Int'l Workshop Peer-to-Peer Systems (IPTPS '06)*, Feb. 2006.
- [27] A. Srinivasan, J. Teitelbaum, and J. Wu, "DRBTS: Distributed Reputation-Based Beacon Trust System," *Proc. Second IEEE Int'l Symp. Dependable, Autonomic and Secure Computing (DASC)*, 2006.
- [28] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," *Proc. 12th Int'l World Wide Web Conf. (WWW '03)*, May 2003.
- [29] F.P. Kelly, "Mathematical Modelling of the Internet," *Mathematics Unlimited—2001 and Beyond*, B. Engquist and W. Schmid, eds., pp. 685-702, Springer, 2001.



Minglu Li received the degree from the School of Electronic Technology, University of Information Engineering, in 1985 and the PhD degree in computer software from Shanghai Jiao Tong University (SJTU), Shanghai, in 1996. He is a full professor and the vice chair of the Department of Computer Science and Engineering and the director of the Grid Computing Center at SJTU. Currently, his research interests include grid computing, services computing, and sensor networks. He has presided over 20 projects supported by the National Natural Science Foundation, National Key Technology R&D Program, 863 Program, 973 Program, and Science and Technology Commission of Shanghai Municipality (STCSM). He has published more than 100 papers in academic journals and international conference proceedings. He is also a member of the Expert Committee of the ChinaGrid Program of the Ministry of Education, a principal scientist of Shanghai-Grid, which is a grand project of STCSM, a member of the Executive Committee of the ChinaGrid Forum, and a member of the Executive Committee of the Technical Committee on Services Computing of the IEEE Computer Society.



Jiadi Yu received the MS degree in computer science from Xi'an Technological University, Xi'an, China, in 2003 and the PhD degree in computer science from Shanghai Jiao Tong University, Shanghai, in 2007. He is a postdoc in the Ad Hoc Network and Network Computing Center and the Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interests include networking, P2P computing, mobile computing, and wireless sensor networks.



Jie Wu is a distinguished professor in the Department of Computer Science and Engineering, Florida Atlantic University, and a program director at the US National Science Foundation. His research interests are in the areas of wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He has published more than 350 papers in various journals and conference proceedings. He is the author of the text *Distributed System Design*, published by the CRC Press. He was on the editorial board of the *IEEE Transactions on Parallel and Distributed Systems* and was a co-guest editor of *Computer* and the *Journal of Parallel and Distributed Computing*. He served as a program cochair for MASS 2004, a program vice chair for ICDCS 2001, and a program vice chair for ICPP 2000. He was also the general chair for MASS 2006 and is the general chair for IPDPS 2008. He was also the recipient of the 1996-1997, 2001-2002, and 2005-2006 Researcher of the Year Award at Florida Atlantic University. He has served as an IEEE Computer Society distinguished visitor and is the chairman of IEEE Technical Committee on Distributed Processing (TCDP). He is a senior member of the IEEE and a member of the ACM.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.