Analyzing the Performance and Fairness of BitTorrent-like Networks Using a General Fluid Model

Yao Yue and Chuang Lin
Department of Computer Science and Technology
Tsinghua University, China
{yueyao, clin}@csnet1.cs.tsinghua.edu.cn

Zhangxi Tan
Department of Electrical Engineering
and Computer Science
University of California, Berkeley, USA
xtan@cs.berkeley.edu

Abstract— **In this paper, a general fluid model is developed to study the performance and fairness of BitTorrent-like networks. The fluid model incorporates two important features, user settings with multiple groups and inter-group data exchange, in a synthesized way to obtain statistics about the system performance. Our numerical results point out some key parameters of the system, such as the staying time of seeders. Generally, selfish behavior does not receive equal performance degradation, and in some scenarios users have strong incentives of free-riding. We also find content delivery can be greatly deterred when malicious free-riders are overwhelming.**

I. Introduction

Peer-to-Peer (P2P) applications have become a major source of Internet traffic and are still on the rise [1]. As a typical P2P software and protocol, BitTorrent [2] has achieved a remarkable success, taking over half of the P2P traffic in some areas [3]. A BitTorrent session is the procedure of distributing a file to interested clients, called *peers*. Regarding BitTorrent sessions, four issues have been widely discussed. The first is scalability, namely, how BitTorrent handles varied population. The second is the maintenance of a session. The third issue is the efficiency, i.e., how fast peers can finish downloading and how well their bandwidths are utilized. The fourth issue concerning BitTorrent is how well it deters selfish behavior, known as free-riding in most P2P literature. Due to the cooperative nature of BitTorrent-like networks, performance and fairness are interwoven and thus should not be treated separately.

Previous works on BitTorrent have taken either an experimental or a theoretical approach. [4] collected realistic data from the tracker's perspective. [5] analyzed a peer's log files after participating in 12 BitTorrent sessions. [6] simulated and altered some major mechanisms in BitTorrent. On the other hand, in [7] a branching process model was used to estimate the service capacity of BitTorrent-like networks after startup. [8] introduced a simple fluid model of identical users, and obtained some analytical solutions as well as the equilibrium strategy for peers. However, performance models in [7], [8] failed to capture the influence of multiple groups and optimistic unchoking.

In this paper, we use a general fluid model to combine a multi-group setting and the choking algorithm. Multiple groups are necessary because they reflect realistic peer composition. The multi-group setting further enables and necessitates the study of *optimistic unchoking*, a mechanism via which different groups unavoidably exchange data. Numerical solutions can be generated for the model to reveal parameters essential to system performance and fairness. Our results show that group composition and the number of seeders are both important to performance. In particular, normal peers may get harmed if the session is dominated by malicious free-riders. Calculation also proves that peers with a smaller uploading capacity do not suffer an equally prolonged downloading time. For certain scenarios, the low-bandwidth peers will have strong incentive of free-riding even if they are performance-sensitive.

The rest of the paper is organized as follows. Section II provides a very brief description of the BitTorrent protocol. Section III describes a general fluid model through three steps. First multiple groups are introduced in a fluid model without data exchange between groups, followed by a discussion of its steady state. Secondly, assuming optimistic unchoking is running, download speed statistics of steady state and convergence time are computed for each group. Thirdly, the steady state of the fluid model is revised to consider the influence of optimistic unchoking. Section IV presents numerical results of the general mode and carries some discussion about the key elements toward better efficiency and fairness. Section V concludes the paper.

II. Background

In BitTorrent, peers aiming at the same file form a large pool called a *swarm*, made up of both *leechers* (downloading peers) and *seeders* (uploading peers). A peer typically keeps connections with at most 40 to 100 peers, which become its *neighbors*. But the number of leechers a peer actually uploads to at one time is only a few. The local peer holds back uploading to most neighbors by marking them as *choked*, otherwise *unchoked*. The leechers actually receiving data are called *downloaders* by the local peer.

To get a complete file, peers join the corresponding session after downloading a small *.torrent* file containing session information. The actual file distributed via BitTorrent is split into much smaller, verifiable parts called pieces. Therefore new leechers can start exchanging data with other peers almost immediately.

The *choking algorithm* decides the choking state of a peer's neighbors, and it consists of *regular unchoking* and optimistic unchoking. Regular unchoking periodically selects neighbors who are offering the best uploading rates. On the other hand, the local leecher tentatively unchokes other neighbors to discover better targets, using optimistic unchoking. In practice, it transfers data to some randomly chosen neighbor for 30 seconds, during this period, the regular unchoking at the remote peer may start sending data in return. Seeders behave quite differently from leechers, because they require no reciprocal uploading and merely consider leechers' downloading capacity. The latest choking algorithm distributes a seeder's uploading chances rather randomly among its neighbors.

III. A General Fluid Model

This section develops a general fluid model. Here we first introduce multiple leecher groups into the model without intergroup data exchange. Then after discussing the impact of optimistic unchoking in detail, the fluid model is revised to take that into account.

A. Introducing Multiple Groups

In a real BitTorrent session seldom do all peers possess the same physical bandwidths, and peers can effortlessly set bandwidth limits in many BitTorrent clients. Therefore, we introduce multiple groups and extend the uni-group fluid model in [8]. In the general model, a set of parameters characterizes a peer group, and only finite groups exist. The file size of each session is defined as 1, and peers' bandwidths can be normalized accordingly. For a swarm of n peer groups, variables and parameters describing the model at time t are listed as following.

λ_i arriving rate of G_i, arrivals follow Poisson distribution.
θ_i the rate a G_i leecher aborts the session.
γ the rate seeders leave a session. Since leaving is manually done, we assume all groups have the same value.
μ_i uploading bandwidth limit of a G_i leecher.
c_i downloading bandwidth limit of a G_i leecher, $c_i \geq \mu_i$.
$x_i(t)$ number of leechers that belong to group G_i at time t.
$y(t)$ number of seeders at time t.
$\mu(t)$ average uploading bandwidth of seeders.
η file sharing efficiency, defined as the probability a leecher has at least one piece requested by its neighbors. [8] argues η is very close to 1 in BitTorrent.

The variables $x_i(t)$, $y(t)$, $\mu(t)$ are decided by the parameters λ_i, θ_i, γ, μ_i, c_i, η. At time t, the size of leecher group i changes by the margin between the arriving rate λ_i and departing rate, which is the sum of abandoning rate $\theta_i x_i$ and the rate group i peers become seeders $min\left\{\mu_i \eta x_i(t) + \mu\rho_i(t)y(t), c_i x_i(t)\right\}$. The size and composition of the seeder group evolve similarly. $\mu(t)$ changes according to the composition of seeders. Hence we obtain a set of differential equations:

$$\begin{aligned}
\frac{\mathrm{d}x_i}{\mathrm{d}t} &= \lambda_i - \theta_i x_i(t) - min\left\{\mu_i \eta x_i(t) + \mu\rho_i(t)y(t), c_i x_i(t)\right\} \\
\frac{\mathrm{d}y}{\mathrm{d}t} &= \sum_{i=1}^{n} min\left\{\mu_i \eta x_i(t) + \mu\rho_i(t)y(t), c_i x_i(t)\right\} - \gamma y(t) \\
\frac{\mathrm{d}\mu(t)}{\mathrm{d}t} &= \sum_{i=1}^{n} min\left\{\frac{\mu_i \eta x_i(t)}{y(t)} + \mu\rho_i(t), \frac{c_i x_i(t)}{y(t)}\right\}\mu_i - \gamma\mu(t)
\end{aligned} \tag{1}$$

The auxiliary variables in Equation (1) are defined as:

$x(t)$ total number of leechers at time t, $x(t) = \sum_{i=1}^{n} \mu_i x_i(t)$.
$\rho_i(t)$ proportion of group i in all leechers, $\rho_i(t) = \frac{x_i(t)}{x(t)}$.

1) Steady State: A BitTorrent session usually has a lasting steady state [4]. Steady state is solved by letting $\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = \frac{\mathrm{d}y(t)}{\mathrm{d}t} = 0$ in Equation (1). By applying iteration to the equations, steady state with any number of groups can be solved.

But here we focus a swarm of two groups, the simplest case reflecting peer disparity and fairness problem. Another benefit is that analytical solutions are available in this case. The steady state value of any variable $a(t)$ is represented as $\bar{a}$, e.g., $\bar{x}_i$, $\bar{y}$. In one situation, all leechers are limited by their uploading bandwidths, namely $\mu_i \eta \bar{x}_i + \bar{\mu}\bar{\rho}_i\bar{y} < c_i \bar{x}_i$ for $i = 1, 2$. Since lack of uploading bandwidth is common, this case should be particularly important. The corresponding steady state equations are:

$$\begin{aligned}
0 &= \lambda_1 - \theta_1 \bar{\rho}\bar{x} - (\mu_1 \eta \bar{\rho}_1 \bar{x} + \bar{\mu}\bar{\rho}_1\bar{y}) \\
0 &= \lambda_2 - \theta_2 \bar{\rho}_2 \bar{x} - (\mu_2 \eta \bar{\rho}_2 \bar{x} + \bar{\mu}\bar{\rho}_2\bar{y}) \\
0 &= \mu_1 \eta \bar{\rho}_1 \bar{x} + \mu_2 \eta \bar{\rho}_2 \bar{x} + \bar{\mu}\bar{y} - \gamma\bar{y} \\
&\bar{\mu}[\mu_1 \eta \bar{\rho}_1 \bar{x} + \mu_2 \eta \bar{\rho}_2 \bar{x} + \bar{\mu}\bar{y}] \\
&\quad = \mu_1(\mu_1 \eta \bar{\rho}_1 \bar{x} + \bar{\mu}\bar{\rho}_1\bar{y}) + \mu_2(\mu_2 \eta \bar{\rho}_2 \bar{x} + \bar{\mu}\bar{\rho}_2\bar{y})
\end{aligned} \tag{2}$$

As mentioned before, $\eta = 1$ can be treated as 1, then $\bar{\rho}_1$ $(0 < \bar{\rho}_1 < 1)$ turns out to be the root of this quadratic equation:

$$\begin{aligned}
0 = \; & \lambda_1 \left(-\mu_2\theta_2 + \gamma\,\mu_2 + \gamma\,\theta_2\right) \\
& + \left[\lambda_1 \left(2\,\mu_2\theta_2 - \gamma\,\mu_2 - \gamma\,\theta_2 - \mu_1\mu_2 - \mu_1\theta_2 + {\mu_1}^2\right)\right. \\
& \quad \left. + \; \lambda_2 \left(\mu_2\theta_1 + \mu_1\mu_2 - \gamma\,\mu_1 - {\mu_2}^2 - \gamma\,\theta_1\right)\right] z \\
& + \left[\lambda_1 \left(\mu_1\theta_2 - \mu_2\theta_2 + \mu_1\mu_2 - {\mu_1}^2\right)\right. \\
& \quad \left. + \; \lambda_2 \left(\mu_1\theta_1 + {\mu_2}^2 - \mu_2\theta_1 - \mu_1\mu_2\right)\right] z^2
\end{aligned} \tag{3}$$

$\bar{x}$, $\bar{y}$ can then expressed as functions of $\bar{\rho}_1$. Of course, when bottlenecks are located elsewhere, the model can be solved using the same method, but easier.

B. The Influence of Optimistic Unchoking

Choking algorithm serves to encourage and maintain cooperation. Choking does not matter for a uni-group swarm, but it plays an important role among multiple groups. While regular unchoking seems straightforward, optimistic unchoking affects the performance and fairness in contradictory ways. On the up side, it offers the dynamic toward the Nash equilibrium state proved in [8]. On the down side, such uploading can be taken advantage of by malicious peers or free-riders.

In this section we calculate a leecher's average downloading rates at steady state and convergence times toward it. In a two-group swarm, it is assumed new leechers only know the existence of the two groups, G_1 with a higher uploading bandwidth and G_2 with a lower one. Leechers also believe each group has over n_u (number of regular unchoking neighbors) members so they take the Nash equilibrium strategy. In addition, a peer distributes its uploading bandwidth equally among downloaders.

1) Converging Time: Now we answers a question: how long does it take a peer to reach steady state? Obviously, only G_1 peers need to be discussed in a two-group case. An optimistic unchoking round is long enough for regular unchoking to take effect between two peers, so we measure how many rounds are needed. To facilitate a uniform discussion, the bidirectional procedure is equated with a double-paced one from a local view. Since the regular unchoking list of a G_1 peer should stabilize after identifying n_u other G_1 leechers, the probability a peer stabilizes right after the k-th round is:

$$\begin{aligned}\mathbb{P}_{n_u}\{R=k\} &= \mathbb{P}_{n_u-1}\{R<k\} \times \mathrm{P}\{G_1 \text{ peer, round } k\}\\ &= f(n_u; N-1, N_1-1, k)\,\frac{n_u}{k}\end{aligned} \tag{4}$$

Function f follows Hypergeometric distribution. Round expectation is:

$$E[R] = n_u \binom{N_1-1}{n_u} \sum_{k=n_u}^{N-N_1+n_u} \frac{\binom{N-N_1}{k-n_u}}{\binom{N-1}{k}} \tag{5}$$

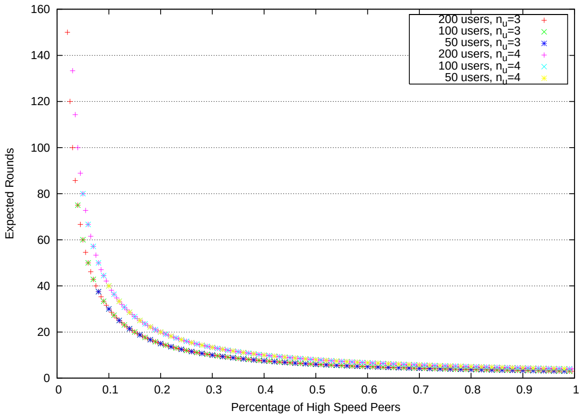


Fig. 1. Convergence Time toward Steady State

We plot the relationship between $E[R]$ and G_1's portion among leechers in Figure 1, and have several discoveries. First, swarm population has almost no influence on expectation, but large swarms help G_1 peers to avoid being trapped by G_2 for long. Second, both expectation and variance of R drop quickly as G_1's portion increases. If G_1 leechers are not rare (say, below 10% in the swarm), they can reach steady state reasonably fast. Third, the convergence time increases no more slowly than n_u. Figure 2 says the difference in expectation between $n_u = 3$ and $n_u = 4$ is roughly 30%, and the difference in standard deviation is below than 20%. These equations and figures also explain the *warming up* period often reported by BitTorrent users.

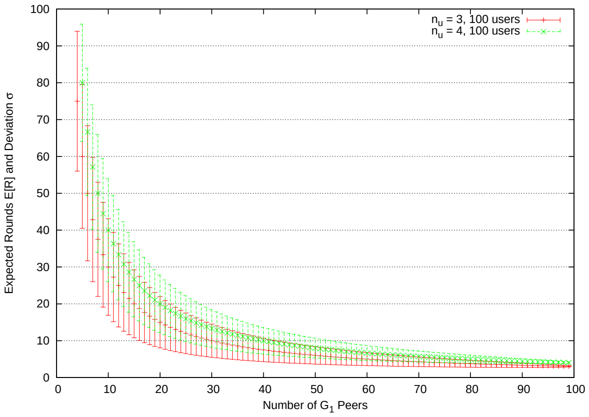


Fig. 2. Expectation and Standard Deviation of Convergence Time

2) Average Downloading Rates: Now we discuss a peer's expected downloading rate from other leechers in steady state. Let $\|G_1\| = N_1$, $\|G_2\| = N_2$, $N_1 + N_2 = N$. The average individual downloading rates of G_1 and G_2, υ_1 and υ_2 are:

$$\begin{aligned}\upsilon_1 &= \mu_1 - \frac{\mu_1-\mu_2}{n_u+1}\,\frac{N_2}{N-n_u-1}\\ \upsilon_2 &= \mu_2 + \frac{\mu_1-\mu_2}{n_u+1}\,\frac{N_1}{N-n_u-1}\end{aligned} \tag{6}$$

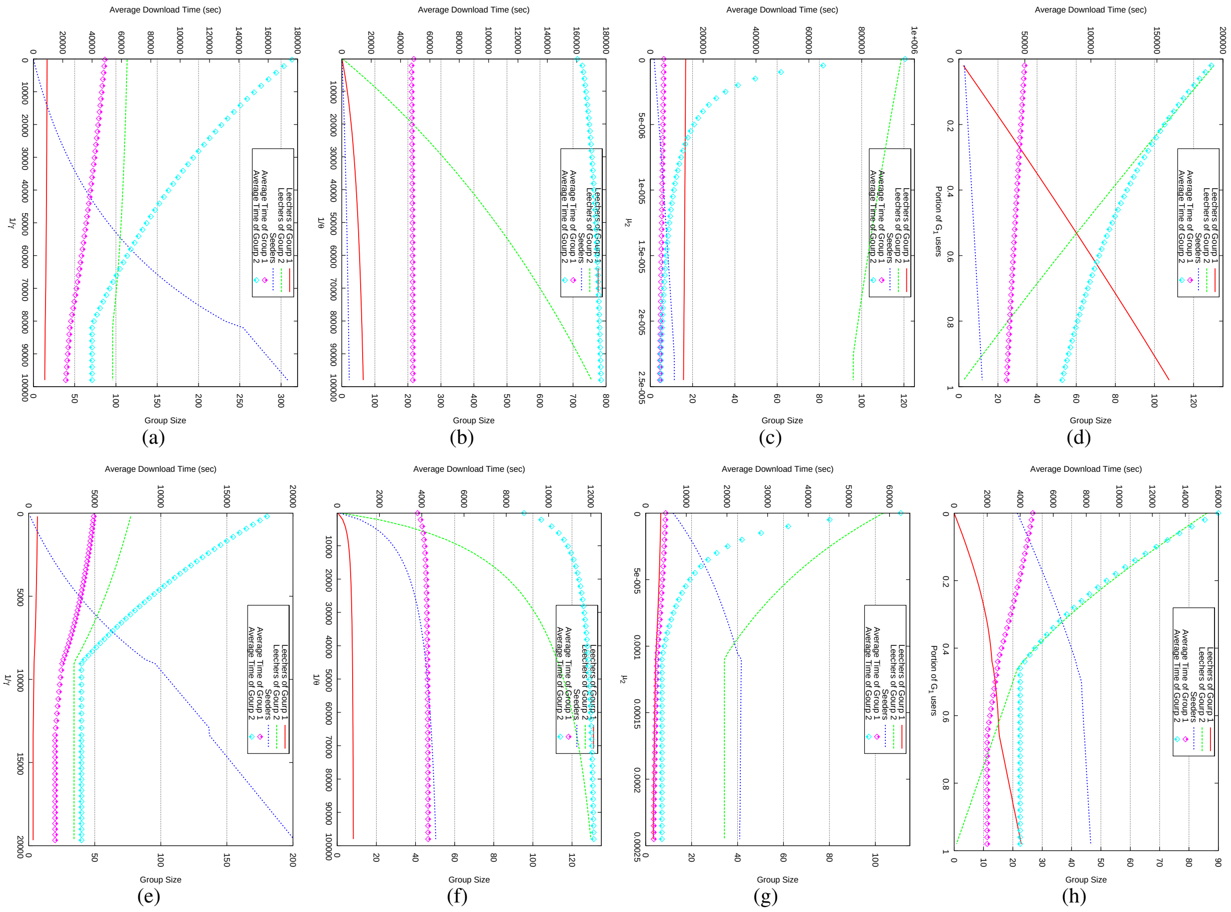
Equation (6) tells us the downloading rate of G_1 peer is lower than its uploading bandwidth. On the contrary, a G_2 leecher will get some extra gain. The absolute difference between the downloading and uploading rate changes linearly according to G_1's relative population.

G_1 peers undergo at most a $\frac{1}{n_u+1}$ discount (compared to its uploading bandwidth) in downloading rate. However, the relative gain of a G_2 peer may vary greatly. When the bandwidth gap is large, G_2 peers' downloading bandwidth can be saturated merely by being optimistic-unchoked.

C. A General Fluid Model

To count in optimistic unchoking, we revise the steady state solution of the fluid model in Section III-A. Previously, Section III-B.1 demonstrates that data exchange through the connection lasts much longer compared with the time spent on finding a peer. Therefore, we believe solving steady state equations, which disregards the impact of swarm dynamics, still offers a decent estimation of the overall situation.

First, the steady state solution of Equation (2 is revisited. Each group's downloading rate from other leechers, i.e. μ_i, is now replaced by $\nu_i(\bar{x}_i)$ from Equation (6). The rate that group i leechers become seeders now becomes $\nu_i \eta \bar{\rho}_i \bar{x} + \bar{\mu} \bar{\rho}_1 \bar{y}$, $i = 1, 2$. And the constraints on system performance bottleneck should be $\nu_i \eta \bar{x}_i + \bar{\mu} \bar{\rho}_i \bar{y} < c_i \bar{x}_i$, $i = 1, 2$. The equations when one or two of G_1's and G_2's downloading bandwidths are saturated are straightforward. We will resort to numerical methods to visualize statistics in the next section.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3. Change of Download Speed and Group Size against Different Parameters

IV. Numerical Results and Analysis

The rich numerical data illustrate how a BitTorrent session is influenced by system parameters, and when occasions occur that free-riding becomes appealing. We validate our calculations with a handful of available real life measurements and simulations.

When applying numerical analysis on Equation (1), we center around two baseline settings in Table I. The trend of average downloading time and group sizes are visualized in Figure 3. The baselines are picked referring to the real-life statistics in [4] as: $\lambda_1 = 0.002$, $\lambda_2 = 0.012$, $\theta_1 = \theta_2 = 0.0001$, $\gamma = 0.00025$ and $\eta = 1$. Two different settings of bandwidths in Table I reflect different file sizes. The relation between uploading and downloading bandwidth agrees with realistic scenarios, where higher-end users usually have more symmetric bandwidths. In Figure 3, subfigures on the first line adopt setting 1, while the rest adopt setting 2.

TABLE I

Default Values of Bandwidth Parameters

parameter	μ_1	c_1	μ_2	c_2
setting 1	0.000025	0.00005	0.000005	0.000025
setting 2	0.00025	0.0005	0.00005	0.00025

A. Performance Analysis

Swarm population is the key for a session to survive, especially seeder population. Calculation other than in Figure 3 proves both leecher and seeder populations grow almost proportional to the arriving rate. On the other side, group sizes are negatively correlated to the abandoning rate and G_2 grows much faster than G_1 as this rate drops. Figure 3(a) and Figure 3(e) show that the average seeder number quickly approaches 0 when seeders' staying time decreases. But a higher μ_2 or higher percentage of G_1 peers makes the session more survivable. In addition, file size plays an important role. Very large files such as in Figure 3(a) to 3(d) tend to keep more leechers but less seeders than smaller files. Statistics of 12 sessions in [5] demonstrates smaller files typically correspond to higher seeder/leecher ratio.

Some interesting features have been discovered about the efficiency. G_1 users are insensitive to the abandoning rate. However, it is not the case with G_2 peers. In setting 2, a near 40% delay is observed when most G_2 users stick around longer. That means the many unfinished downloads (as reported by [4]) actually promote the performance of remaining peers. The reason is that a higher abandoning rate leads to more G_1 peers within the swarm. On the other hand, the number of seeders is significant to the efficiency of all. In setting 2, when the seeders' average staying time increases from 1,000 seconds to 10,000 seconds, downloading time of a G_1 and G_2 leecher shortens by 50% and 76% respectively. The evolution concerning μ_2 is worth some special attention here, because Figure 3(c) and Figure 3(g) show that allowing a slightly more than zero uploading greatly reduce the downloading time for G_2 peers. This means holding back moderate bandwidths leads to acceptable downloading rates. We will return to this phenomenon in Section IV-B.

To find out the scalability of BitTorrent, several other settings have been computed by multiplying all λ_i by a factor. BitTorrent scales well except under some extreme conditions. For large swarms, bandwidth mismatch in regular unchoking rarely occurs. However, the choking algorithm says if the G_1 population is less than n_u, a G_1 peer has to regular-unchoke one or more G_2 peers. In this sense, big swarms, or swarms with high arriving rates and low leaving rates, have more predictable performance.

B. Analysis of the Fairness Problem

First let us take a close look at μ_2, and we find in Figure 3(c), a G_2 leecher' providing 60% the uploading bandwidth of G_1 actually downloads at 62% the rate of G_1. Under setting 2, a G_2 peer only needs 40% G_1's uploading bandwidth to receive a 59% relative downloading rate. Table II lists the thresholds at which peers' downloading bandwidths become saturated. In both settings, G_2 peers reach saturation before G_1. Besides, it is more likely to achieve full-speed download with smaller files, mainly because of the abundant seeders.

TABLE II
THRESHOLDS OF FULL-CAPACITY DOWNLOADING

parameters	setting 1		setting 2	
	G_1	G_2	G_1	G_2
γ	-	$< 1.22e-5$	$< 7.49e-5$	$< 1.11e-4$
μ_2	-	$> 2.26e-5$	$> 2.31e-4$	$> 1.08e-4$
G_1 (%)	-	-	> 65.5	> 45.2

"-" means such threshold does not exist.

The combination of optimistic unchoking and seeders' unselfish uploading makes free-riding more attractive. Under default setting 1 (setting 2), G_1 leechers take 21.7% (10.9%) more time in the general fluid model compared with in a model without inter-group data exchange. From G_2 peers' view point, with optimistic unchoking they save 10.3% (4.4%) downloading time. Meanwhile, without optimistic unchoking the bandwidth saturation becomes less likely for G_2 but easier for G_1. Based on a similar observation in simulation, a bandwidth estimation method is suggested in [6] to replace optimistic unchoking.

C. Discussion

i) Security: After demonstrating BitTorrent is quite scalable facing flash arrivals of normal peer composition, it remains doubtful if it can resists abundant malicious free-riders. Our convergence time of choking algorithm in Section III-B.1 shows with very low percentage of high-bandwidth peers, they may be easily trapped by low-bandwidth users and find it hard to achieve ideal downloading rates.

ii) Seeder: Long-staying peers can upload much more data as seeders than leechers, and are crucial to maintain a session. What's more, with plenty of seeders, a session with a low utilization rate of uploading bandwidths can still be very efficient. Unfortunately, the current BitTorrent protocol lacks incentives to keep seeders around. Consequently, performance of BitTorrent sessions can be quite unstable even for the same peer composition. We notice that there is a distinct relationship between file size and seeder population, making sharing smaller files more efficient than larger ones. However, sessions of smaller files tend to cease more easily than larger ones.

iii) Free-riding: BitTorrent is vulnerable to free-riding under certain circumstances. Figure 3 and Table II provide sufficiency conditions of free-riding in a theoretical model. Nevertheless, system performance in BitTorrent is always positively related to free-riding incentives. Therefore, the fairness problem of a normal BitTorrent session, i.e., without a flash-crowd of malicious free-riders, is not critical.

V. CONCLUSION

In this paper, we have studied BitTorrent-like networks using a general fluid model. The emphasis is laid on the steady state performance and fairness of stable sessions. With numerical methods, influences of major system parameters as well as optimistic unchoking are discussed. We find seeders play a key roll in achieving good overall performance, yet there lacks encouraging mechanisms. We also find BitTorrent is not strictly scalable or safe against abundant maliciously selfish leechers. The free-riding incentives in BitTorrent are stronger than previously believed in theoretical study.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (No.90412012) and Microsoft Research Asia Program Enhance Trust of P2P system with Group Signature Technique in Privacy and Security.

REFERENCES

[1] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 219–232, 2004.
[2] [Online]. Available: http://www.bittorrent.com
[3] T. Mennecke. (2005, Jan.) Bittorrent remains powerhouse network. [Online]. Available: http://www.slyck.com/news.php?story=649
[4] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. Felber, A. A. Hamra, and L. Garcs-Erice, "Dissecting bittorrent: Five months in a torrents lifetime," in *Proc. PAM 2004*, Apr. 2004, pp. 1–11.
[5] P. M. Arnaud Legout, Guillaume Urvoy-Keller, "Understanding bittorrent: An experimental perspective," INRIA Sophia Antipolis & Institut Eurecom, Tech. Rep. inria-00000156, version 3, Nov. 2005.
[6] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and improving bittorrent performance," Carnegie Mellon University &Microsoft Research, Tech. Rep. MSR-TR-2005-03, Feb. 2005.
[7] X. Yang and G. de Veciana, "Service capacity of peer to peer networks," in *INFOCOM '04*, 2004, pp. 2242–2252.
[8] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *SIGCOMM '04*, 2004, pp. 367–378.