BitTorrent's Dilemma: Enhancing Reciprocity or Reducing Inequity

Jia, AL; d' Acunto, L; Meulpolder, M; Pouwelse, JA; Epema, DHJ

DOI
10.1109/CCNC.2011.5766581

Publication date
2011

Document Version
Accepted author manuscript

Published in
IEEE Consumer Communications and Networking Conference (CCNC '11)

Citation (APA)
Jia, AL., d' Acunto, L., Meulpolder, M., Pouwelse, JA., & Epema, DHJ. (2011). BitTorrent's Dilemma: Enhancing Reciprocity or Reducing Inequity. In M. Kolberg (Ed.), *IEEE Consumer Communications and Networking Conference (CCNC '11)* (pp. 705-709). IEEE Society. https://doi.org/10.1109/CCNC.2011.5766581

BitTorrent's Dilemma: Enhancing Reciprocity or Reducing Inequity

Adele L. Jia, Lucia D'Acunto, Michel Meulpolder, Johan A. Pouwelse and Dick H.J. Epema
Parallel and Distributed Systems Group
Department of Computer Science, Delft University of Technology, the Netherlands
Email: adele.lu.jia@gmail.com

***Abstract*—Enhancing reciprocity has been one of the primary motivations for the design of incentive policies in BitTorrent-like P2P systems. Reciprocity implies that peers need to contribute their bandwidth to other peers if they want to receive bandwidth in return. However, the over-provisioning that characterizes today's BitTorrent communities and the development of many next-generation P2P systems with real-time constraints (e.g., for live and on-demand streaming) suggest that more effort can be devoted to reducing the inequity (i.e., the difference of service received) among peers, rather than only enhancing reciprocity. Inspired by this observation, in this work we analyze in detail several incentive mechanisms that are used in BitTorrent systems, and explore several strategies that influence the balance between reciprocity and equity. Our study shows that (i) reducing inequity leads to a better overall system performance, and (ii) the behavior of seeders (i.e., peers that hold a complete copy of the file and upload it for free) influences whether reciprocity is enhanced or inequity reduced.**

I. Introduction

BitTorrent is a popular peer-to-peer (P2P) protocol for file distribution over the Internet. In order to induce cooperation among peers, BitTorrent incorporates an incentive mechanism based on direct *reciprocity*, where nodes prefer uploading to peers who have contributed to them in the past at the highest speeds. This incentive mechanism was designed to allow peers to obtain their file of interest even in *resource-constrained* scenarios, e.g., when only a few peers exist that hold a complete copy of the file (*seeders*, in BitTorrent terminology), or during flash-crowds.

However, the *BitTorrent ecosystem* is nowadays extremely diverse. For example, a recent measurement study [8] has shown that most BitTorrent communities are over-provisioned, i.e., there are significantly more seeders than downloaders. Also, the design of many next-generation P2P systems, such as those for the distribution of live and on-demand streaming [2] [4] [5], has been inspired by the BitTorrent paradigm. The real-time constraints of these systems require that all peers are provided with a certain minimum download speed (in order to support the bitrate of the video) and that peers do not earn more utility in downloading at rates much faster than that. These observations suggest that it is not necessary to always enhance reciprocity; in some cases it is more advisable to reduce *inequity* among peers, instead. One of the first studies of this trade-off in BitTorrent-like systems was provided by Fan *et al.* [3].

In this paper, we extend earlier work by introducing a more detailed model and analyzing *how* the incentive mechanism of the BitTorrent protocol can be tuned to enhance reciprocity or reduce inequity. Furthermore, in our study we consider the implications of exchanging BitTorrent's standard incentive mechanism with one that is based on effort rather than speed. Finally, we also analyze the role of the seeders. Hence, we provide significant more insights in the implications of this important trade-off. Our contributions can be summarized as follows:

- we provide an analytical model that characterizes the inherent relationship between a peer's performance and the design parameters of the BitTorrent protocol that are responsible for its incentive mechanism (Section II-B).
- we use this model to analyze different strategies to enhance reciprocity, reduce inequity and understand the role of the seeders (Section III).
- we consider the impact of these strategies on the overall system performance (Section III).

Overall, our work aids in informing the design choices that best fit the requirements of a BitTorrent-like P2P system.

II. A Fluid Model for BitTorrent

In this section we first introduce the basics of the BitTorrent protocol which are relevant for our work, then we present our model, and finally we illustrate its validation by means of a discrete-event simulator.

A. BitTorrent Overview

Incentive policies play a key role in BitTorrent-like systems, as they determine how peers distribute their limited upload bandwidth to other peers. BitTorrent's original incentive policy is *tit-for-tat* (TFT), in which a peer favors other peers that have recently reciprocated at the highest rate. More specifically, every peer has a number of upload slots available, which are divided into two categories, *regular unchoke slots* and *optimistic unchoke slots*. Downloaders (referred to as *leechers*, in BitTorrent terminology) choose which peers will be allocated to regular unchoke slots according to TFT. On the contrary, peers to be allocated to optimistic unchoke slots are chosen randomly from the neighbors set. While regular unchoke slots are used to enhance reciprocity, optimistic unchoke slots serve the purpose of 1) potentially discovering new faster peers and

2) allowing new peers to bootstrap (i.e., obtain their first pieces of the file).

BitTorrent systems also include special peers called seeders, who have a complete copy of the file and share it without any direct benefit to do so. Two popular seeding policies are: 1) *favoring fast peers* (FF): seeders allocate their regular upload slots to peers that downloaded at the fastest rates and optimistic unchoke slots randomly; 2) *random seeding* (RS): seeders have no preference and just choose peers randomly.

Notation	Definition
F	the size of the file shared in the swarm.
μ_i	the upload capacity of a peer in class i.
D_i	the download capacity of a peer in class i.
d_i	the per connection download capacity of a peer in class i.
u_i	number of unchoke slots opened by a peer in class i, $u_i^{(reg)}$ and $u_i^{(op)}$ for regular and optimistic unchoke slot.
x_i	number of leechers in class i.
π_i	fraction of leechers in class i, $\pi_i = x_i / \sum_i x_i$.
y_i	number of seeders in class i.
λ_i	the arrival rate of leechers in class i.
γ_i	the rate at which seeders in class i leave the system.
α_{ij}	the number of upload slots allocated by a leecher in class i to a leecher in class j.
β_{ij}	the number of upload slots allocated by a seeder in class i to a leecher in class j.
n_i	the number of download slots opened by a class i leecher
U_{ij}	the total upload bandwidth allocated from class i to class j.
D_{ij}	the fraction of upload capacity of leechers in class i allocated to leechers in class j.
S_{ij}	the fraction of upload capacity of seeders in class i allocated to leechers in class j.

TABLE I
NOTATION OF OUR BITTORRENT MODEL

B. Model description

We follow a similar fluid modeling approach as Qiu *et al.* [9] and Meulpolder *et al.* [7]. The notation we use is shown in Table I. Similar to the approach in [7], we group peers into different classes according to their upload capacities, but we introduce the notion of *per connection download capacity*. For each class i, the evolution of the number of leechers, $x_i(t)$, and the number of seeders, $y_i(t)$, is as follows:

$$\begin{aligned} \frac{dx_i(t)}{dt} &= \lambda_i - \frac{\sum_j U_{ji}(t)}{F}, \\ \frac{dy_i(t)}{dt} &= \frac{\sum_j U_{ji}(t)}{F} - \gamma_i y_i(t). \end{aligned} \tag{1}$$

In a steady state, although peers are arriving and departing, the total system population is constant. So it holds that $\frac{dx_i(t)}{dt} = \frac{dy_i(t)}{dt} \equiv 0$, which implies:

$$\lambda_i F = \gamma_i y_i F = \sum_j U_{ji} = \sum_j (D_{ji} x_j + S_{ji} y_j)\mu_j. \tag{2}$$

Combining this with Little's Law ($x_i = \lambda_i T_i$), the average download speed for leechers in class i can be calculated as:

$$\frac{F}{T_i} = \frac{F\lambda_i}{x_i} = \frac{1}{x_i}\sum_j (D_{ji} x_j + S_{ji} y_j)\mu_j. \tag{3}$$

We discuss how to derive the upload bandwidth allocation (D_{ji} and S_{ji} respectively) in the following subsection.

C. Bandwidth allocation

Without loss of generality, we assume that $\mu_1 < \mu_2 < ... < \mu_N$ and $D_1 < D_2 < ... < D_N$.

Leechers utilize the TFT policy. As an indirect result, high capacity peers only unchoke low capacity peers using optimistic unchoke slots:

$$\alpha_{ij} = u_i^{(op)} \pi_j \quad i, j = 1, 2, ..., N, i > j. \tag{4}$$

Due to their faster upload speed, higher-capacity leechers will get reciprocated when they upload to lower-capacity leechers. On average, each leecher in class j should reciprocate $(\alpha_{ij} x_i)/x_j = u_i^{(op)} \pi_i$ leechers in class i, as long as it has enough upload slots. In case there are not enough upload slots, leechers in higher classes are reciprocated first, i.e.:

$$\alpha_{ji} = \min\{u_i^{(op)} \pi_i, u_j^{(reg)} - \sum_{i<p\leq N} \alpha_{jp}\} + u_j^{(op)} \pi_i. \tag{5}$$

Seeders do not need to be reciprocated since they only upload altruistically. For seeders who adopt the FF policy we have:

$$\begin{aligned} \beta_{iN} &= u_i^{(reg)} + u_i^{(op)} \pi_N, \\ \beta_{ij} &= u_i^{(op)} \pi_j \quad \forall i, j \ and \ j < N, \end{aligned} \tag{6}$$

while for seeders who adopt the RS policy it holds:

$$\beta_{ij} = u_i \pi_j. \tag{7}$$

BitTorrent uses TCP as transport layer protocol. TCP specifies that a peer's upload (download) capacity is equally divided over all connections, unless some of the connections have a bottleneck. When such a bottleneck exists, normally the leftover bandwidth is equally divided over other connections with a higher link capacity. Taking this into account and the fact that, in a steady state, peers in the same class receive a similar service, the average number of download connections and the per connection download capacity for a peer in class i can be calculated as:

$$\begin{aligned} n_i &= \frac{\sum_{1\leq j\leq N} \alpha_{ji} x_j + \beta_{ji} y_j}{x_i}, \\ d_i &= \frac{D_i}{n_i}. \end{aligned} \tag{8}$$

We now reorder the leechers according to d_i, and we assume that $d_1 < d_2 < ... < d_N$. The bandwidth allocation can be calculated as:

$$D_{ij} = \frac{\min\{\frac{\mu_i(1-\sum_{p<j} D_{ip})}{\sum_{k\geq j} \alpha_{ik}}, d_j\} \cdot \alpha_{ij}}{\mu_i}. \tag{9}$$

Replacing D_{ij}, α_{ij} with S_{ij}, β_{ij} respectively, we can calculate a seeder's upload bandwidth allocation in a similar way.

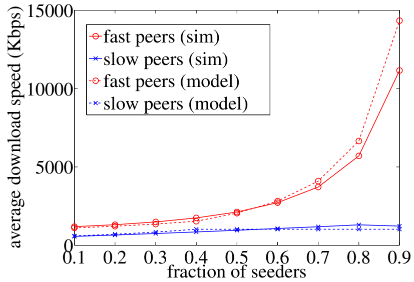


Fig. 1. The average download speeds of fast and slow peers in a system with 50 fast peers and 50 slow peers, for different fraction of seeders. The capacities of peers are the following: 1024 Kbps up and ∞ down for fast peers; 512 Kbps up and 1024 Kbps down for slow peers. Seeders use the FF policy.

D. Model Validation

We have validated our model by means of extensive simulations using a discrete-event simulator that accurately emulates the behavior of BitTorrent at the level of piece transfers. Fig. 1 illustrates the simulation results against the model predictions for a system with two classes of peers, fast and slow, from which we can make the following observations:

1. the model predictions are close to the simulation results;

2. the average download speed of both fast and slow peers increases when there are more seeders;

3. the model predictions become less accurate as the fraction of seeders grows. This can be explained considering that, when a high fraction of peers are seeders (above 70 % in this case), fast leechers have a hard time in finding other fast leechers to reciprocate with. While in our model we assume that, in a steady state, leechers can always find enough other leechers.

III. Analysis

In this section, we analyze the balance between enhancing reciprocity or reducing inequity in BitTorrent. Based on our model, we evaluate the following candidate strategies:

A) fast peers opening more regular unchoke slots;
B) all peers opening more optimistic unchoke slots:
C) replacing TFT with an effort-based incentive policy;
D) seeders' role: favoring fast peers vs seeding randomly.

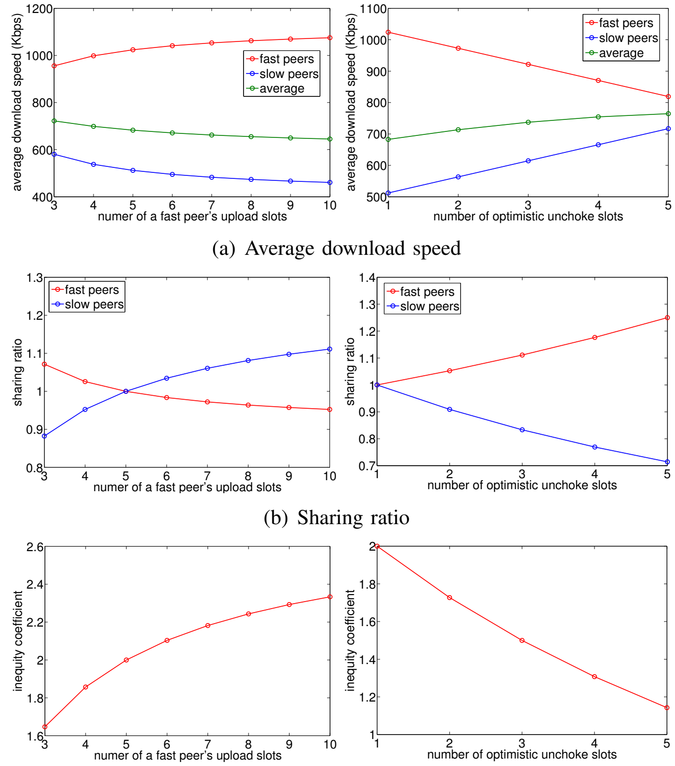
We use the following performance metrics:

1. *download speed*: we use this metric to characterize performance;

2. *sharing ratio*: the ratio between the total amount of data uploaded and downloaded; this metric represents fairness in relation to contribution to the system (e.g., a sharing ratio close to 1 for all peers means that all peers have contributed as much data as they have consumed);

3. *inequity coefficient*: the largest download speed divided by the smallest download speed; it indicates fairness in relation to the bandwidth capacity that peers receive from the system.

Unless stated otherwise, we consider a system with two classes of peers, fast (1024 Kbps up and ∞ down) and slow (512 Kbps up and 1024 down).



(a) Average download speed

(b) Sharing ratio

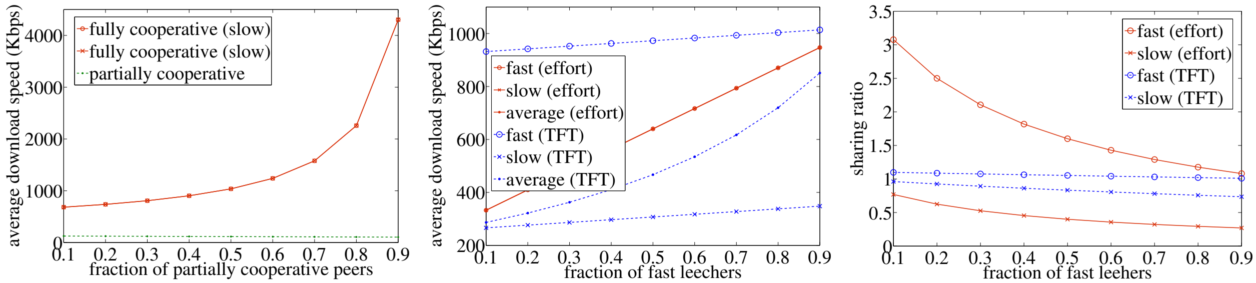
(c) Inequity coefficient

Fig. 2. The influence of the number of upload slot in a system with 100 leechers and no seeders.

A. Strategy 1: enhancing reciprocity with fast leechers opening more regular slots

Regardless of a peer's class, opening more upload slots can help a peer to 1) find more potential fast peers, or to 2) weaken another peer's potential monopoly on its uploading bandwidth since less bandwidth will be allocated to each upload slot. On the other hand, opening too many slots is neither realistic nor reasonable, since too many TCP connections could deteriorate link performance. Also it would become harder for slow peers to succeed in competing for reciprocity with faster peers.

Given the above considerations, fast peers have a stronger motivation to open more slots than slow peers, since they may benefit from more extensive exploration, while remaining competitive in TFT. Having fast peers open more upload slots is a way to enhance reciprocity, as more bandwidth will be allocate to the regular unchoke slots. Fig. 2(a) shows that as the number of upload slots of fast peers increases, their download speed improves (we can observe a growth of 10% when the number of slots goes from 3 to 10), while the average download speed of all peers decreases (10% with the number of slots from 3 to 10). This is due to the increasing inequity (almost 50%) between the two classes of peers, as shown in Fig. 2(c). On the other hand, we notice that the sharing ratio of fast peers decreases as they open more slots, and that of slow peers increases (Fig. 2(b)). The perfect reciprocity (sharing ratio equal to 1 for both fast and slow peers) is achieved when fast peers open 5 upload slots.



(a) Influence of the fraction of partially cooperative peers

(b) Comparison between effort-based and TFT: average download speed

(c) Comparison between effort-based and TFT: sharing ratio

Fig. 3. The performance of effort-based mechanism. The fast and slow peers' upload capacities are 1024 and 256 Kbps, respectively.

In the following theorem we state the conditions necessary to achieve the perfect reciprocity.

Theorem. *In a BitTorrent system with two classes of peers, no seeders, and no download bottleneck, perfect reciprocity is achieved if and only if:*

$$\frac{\mu_f u_s}{\mu_s u_f} = \frac{u_f^{(op)} + u_s^{(op)}}{u_f^{(op)}}. \tag{10}$$

Proof: We first show that for a system with perfect reciprocity, Eq. 10 holds. The sharing ratio of a leecher in class i in a steady state is equal to the ratio of its upload and download speed, i.e.:

$$\frac{\mu_i}{\lambda_i F/x_i} = \frac{\mu_i x_i}{F\sum_{j\in\{f,s\}} D_{ji} x_j \mu_j}. \tag{11}$$

Perfect reciprocity implies that leechers in different classes achieve the same sharing ratio, i.e.:

$$\frac{\mu_f x_f}{\sum_{j\in\{f,s\}} D_{jf} x_j \mu_j} = \frac{\mu_s x_s}{\sum_{j\in\{f,s\}} D_{js} x_j \mu_j}. \tag{12}$$

It follows that Eq. 10 holds.

Next we show that when Eq. 10 holds, perfect reciprocity is achieved. Substituting Eq. 10 into Eq. 11, we get Eq. 12, which implies that fast and slow leechers have the same sharing ratio. It follows that perfect reciprocity is achieved. ∎

From the above theorem it follows that, when we use $\mu_f = 1024, \mu_s = 512, u_s = 5$ and $u_s^{(op)} = u_f^{(op)} = 1$, a perfect reciprocity is obtained for $u_f = u_s = 5$.

B. Strategy 2: reducing inequity with leechers opening more optimistic unchoke slots

Here we analyze the influence of having all peers open more optimistic unchoke slots. While peers always open 5 unchoke slots in total, we let the number of their optimistic unchoke slots vary from 1 to 5. As we can see in Fig. 2(a), in this way the download speed of slow leechers is improved by 40%, at the expense of the fast leechers. Interestingly, the average download speed of the whole population increases of 15%. Moreover, we observe a 45% decrease of the inequity coefficient (Fig. 2(b)).

However, it should be noted that by having peers open more optimistic unchoke slots, the effectiveness of TFT is reduced, as a peer that does not contribute is chosen with the same probability as a cooperative slow peer.

C. Strategy 3: reducing inequity by replacing TFT with effort-based incentives

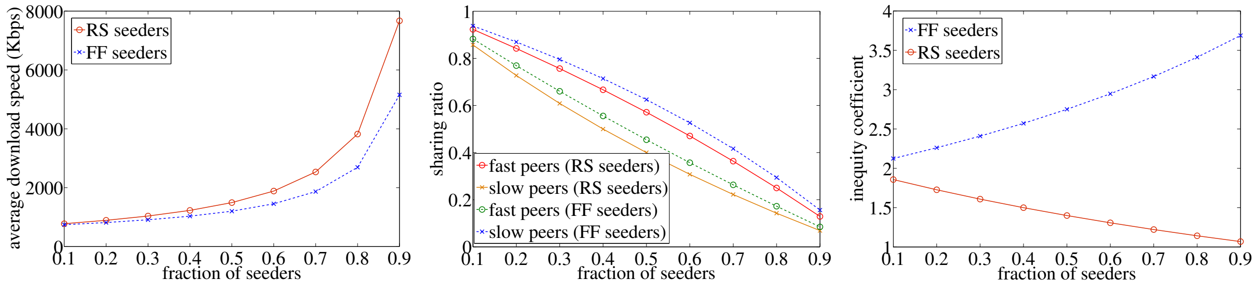
Rahman *et al.* [10] have recently proposed a novel incentive mechanism based on effort, rather than speed. More specifically, peers are not rewarded based on the absolute amount of data they provided, but based on the relative amount of bandwidth they make available (utilized or not). With this approach, a slow peer offering all its bandwidth to the system is preferred over a fast peer offering 0.9 of its total bandwidth.

Consider that there are two types of peers in the system, peers that contribute all their upload bandwidth (*fully cooperative*) and peers that only contribute a fraction of it (*partially cooperative*). Let n_p represent the number of partially cooperative peers, and n_{ff}, n_{fs} represent the number of fully cooperative peers that have a low or high upload capacity respectively. Each peer reciprocates fully cooperative peers by allocating regular unchoke slots to them, and punishes partially cooperative peers by only optimistically unchoking them. The slot allocation for each class of peers can be calculated as:

$$\begin{aligned}
\alpha_{i(p)} &= \frac{u_i^{(op)} n_p}{n_{ff} + n_{fs} + n_p} \\
\alpha_{i(ff)} &= \frac{(u_i - \alpha_{i(p)}) n_{ff}}{n_{ff} + n_{fs}} \\
\alpha_{i(fs)} &= \frac{(u_i - \alpha_{i(p)}) n_{fs}}{n_{ff} + n_{fs}} \qquad \forall i \in \{p, ff, fs\}.
\end{aligned} \tag{13}$$

Given Eq. 13, the upload bandwidth allocation can be calculated in a similar way as in our earlier analysis.

The idea of this incentive scheme is to reduce inequity among the fully cooperative peers while still punishing the partially cooperative peers. Its effectiveness can be observed in Fig. 3(a). In a system where all peers are fully cooperative, the effort-based scheme eliminates the system's inequity and achieves a better overall performance. The average download speed using effort-based incentives is always higher than when using TFT (see Fig. 3(b)). Furthermore, the effort-based



(a) Average download speed (b) Sharing ratio (c) Inequity

Fig. 4. The influence of the seeding strategy: favoring fast peers (FF) or randomly seeding (RS)

mechanism leads to a more equal sharing ratio between fast and slow peers (see Fig. 3(c)).

D. Strategy 4: enhancing reciprocity or reducing equity with a seeder's policies

The mainline BitTorrent client has been implemented with two different seeding strategies in different releases. One is the favoring of fast peers. This strategy accelerates a fast leecher's ability to finish downloading, thereby potentially having it serve as fast seeder in the system sooner. The other strategy is seeding randomly. The first strategy is resource-constrained oriented, as it aims at increasing the serving capacity quickly. The second strategy is more equity oriented, as all peers are treated in the same way.

We have applied our model to analyze and compare these two strategies. Fig. 4(a) and Fig. 4(c) show that if seeders seed randomly, the system achieves a better overall performance (in terms of a higher average download speed) and the inequity is reduced. On the contrary, if seeders favor fast peers, the reciprocity is enhanced: both fast and slow peers have a sharing ratio higher than in a system where seeders adopt random seeding (Fig. 4(b)).

IV. Related work

There are a number of studies on modeling and improving BitTorrent's incentive policies. Some earlier work focuses only on homogeneous systems [6], [12], [9]. In [11], the authors consider heterogeneous BitTorrent systems, but only with two classes of peers. Fan *et al.* [3] have developed a general heterogeneous model to evaluate the tradeoff between performance and fairness. Meulpolder *et al.* [7] and Chow *et al.* [1] also provide models for heterogeneous BitTorrent systems, with which they analyze the clustering and data distribution in BitTorrent swarms. While these works all focus on a particular design, we analyze the performance of different incentive policies from a higher level: we consider different BitTorrent applications and stress that merely enhancing reciprocity is not sufficient in the design of a good incentive policy. We furthermore identify several strategies that can be used to enhance reciprocity or reduce inequity.

V. Conclusions

In this work, we have provided an analytical model for heterogeneous BitTorrent systems that captures the essence of BitTorrent's incentive policy. Based on our model, we have analyzed how TFT could enhance reciprocity or reduce inequity by carefully tuning the number of regular and optimistic unchoke slots. We have also compared TFT to an effort-based incentive policy, and showed that a policy that focuses on reducing inequity leads to a BitTorrent system that achieves a better overall performance. Finally, we have analyzed different seeding policies and our results show that, although seeders do not need to be reciprocated, they can still be used to further enhance reciprocity or reduce inequity among leechers.

Acknowledgements

This work was supported by the Future and Emerging Technologies programme FP7-COSI-ICT of the European Commission through project QLectives (grant no.: 231200) and the EU FP7 project P2PNEXT (grant no.: 216217).

References

[1] A. L. H. Chow, L. Golubchik and V. Nisra. Bittorrent: An extensible heterogeneous model. In *Proceedings of IEEE INFOCOM*, 2009.

[2] A. Vlavianos, M. Iliofotou and M. Faloutsos. BiToS: Enhancing BitTorrent for Supporting Streaming Applications. In *Proceeding of IEEE Global Internet Symposium*, 2006.

[3] B. Fan, D. M. Chiu and J. C. Lui. The delicate tradeoffs in bittorrent-like file sharing protocol design. In *Proceedings of IEEE ICNP*, 2006.

[4] Y. Qu G. Y. Keung C. Lin J. Liu B. Li, S. Xie and X. Zhang. Inside the New Coolstreaming: Principles, Measurements and Performance Implications. In *Proceedings of IEEE INFOCOM*, 2008.

[5] J.J.D. Mol, J. A. Pouwelse, M. Meulpolder, D.H.J. Epema and H.J. Sips. Give-to-Get: Free-riding-resilient Video-on-Demand in P2P Systems. In *Proceeding of SPIE MMCN*, 2008.

[6] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding and X. Zhang. Measurements, analysis, and modeling of bittorrent-like systems. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, 2005.

[7] M. Meulpolder, J.A. Pouwelse, D.H.J. Epema and H.J. Sips. Modeling and Analysis of Bandwidth-Inhomogeneous Swarms in BitTorrent. In *9th International Conference on P2P Systems (IEEE P2P '09)*, 2009.

[8] M. Meulpolder, L. D'Acunto, M. Capota, M. Wojciechowski, J.A. Pouwelse, D.H.J. Epema and H.J. Sips. Public and private bittorrent communities: A measurement study. In *IPTPS*, 2010.

[9] D. Qiu and R. Srikant. Modeling and performance analysis of bit torrent-like peer-to-peer networks. In *ACM SIGCOM*, 2004.

[10] R. Rahman, M. Meulpolder, D. Hales, J.A. Pouwelse, D.H.J. Epema and H.J. Sips. Improving efficiency and fairness in p2p systems with effort-based incentives. In *Proceedings of ICC*, 2010.

[11] W. C. Liao, F. Papadopoulos and K. Psounis. Performance analysis of bittorrent-like systems with heterogeneous users. In *PERFORMANCE '07: Proceedings of the 26th International Symposium on Computer Performance, Modeling, Measurements, and Evaluation*, 2007.

[12] Y. Tian, D. Wu and K. W. Ng. Modeling, analysis and improvement for bittorrent-like file sharing networks. In *Proceedings of INFOCOM*, 2006.