Incentives in BitTorrent Induce Free Riding[*]

Seung Jun
College of Computing
Georgia Institute of Technology
Atlanta, Georgia
jun@cc.gatech.edu

Mustaque Ahamad
College of Computing
Georgia Institute of Technology
Atlanta, Georgia
mustaq@cc.gatech.edu

ABSTRACT

We investigate the incentive mechanism of BitTorrent, which is a peer-to-peer file distribution system. As downloaders in BitTorrent are faced with the conflict between the eagerness to download and the unwillingness to upload, we relate this problem to the iterated prisoner's dilemma, which suggests guidelines to design a good incentive mechanism. Based on these guidelines, we propose a new, simple incentive mechanism. Our analysis and the experimental results using PlanetLab show that the original incentive mechanism of BitTorrent can induce free riding because it is not effective in rewarding and punishing downloaders properly. In contrast, a new mechanism proposed by us is shown to be more robust against free riders.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed Applications*; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Measurement, Economics

Keywords

Data dissemination, BitTorrent, prisoner's dilemma, strategy, incentive mechanisms

1. INTRODUCTION

While cooperation is key to the success of a peer-to-peer system, it is difficult to cultivate without an effective incentive mechanism. In fact, many peer-to-peer systems lack such a mechanism and consequently suffer from free riding [1]. One of the few systems regarded as having an incentive mechanism is BitTorrent (BT), a peer-to-peer file distribution system that has been designed for distributing large files under massive, sudden demand [5]. It is used so widely that it takes up as much as one third of the Internet traffic [14].

In BT, as peers download a file by exchanging the fragments of the file with others, they are faced with the conflict between the eagerness to download and the unwillingness to upload. The desire of a peer to maximize its utility by balancing these apparently opposing goals leads to two related questions. First, from the microscopic view, what is the best strategy that maximizes the utility of an individual peer? Second, from the macroscopic view, how should the incentive mechanism be designed to encourage individuals to cooperate? As suggested above, to emphasize this duality, we refer to a particular way of behaving as a *strategy* from the individual perspective or as an *incentive mechanism* from the global perspective.

In this paper, we investigate the incentive mechanism of BT. Based on the experimental results of running the original code of BT on PlanetLab nodes, we show that free riders finish downloads as early as those who contribute substantially. That is, free riders are not punished properly, and those who substantially contribute are not rewarded appropriately. The lack of reward and punishment can induce free riding to such a degree that the system becomes inefficient. To address this issue, we propose a new mechanism that is robust against free riders and therefore, with this mechanism, the system should be sustainable in the long run. We also provide a game-theoretic framework for understanding the conflict and cooperation between peers. Borrowing lessons from the iterated prisoner's dilemma [2], we present properties shared by good strategies and, based on them, compare the two incentive mechanisms.

The remainder of this paper is organized as follows. Section 2 shows the relationship between BT and the prisoner's dilemma. In Section 3, we explain the incentive mechanism of BT and propose a new mechanism. Both mechanisms are evaluated in Section 4. The related work is presented in Section 5, and the conclusions and the future work are discussed in Section 6.

2. GAME-THEORETIC FRAMEWORK

In this section, we relate BitTorrent to the iterated prisoner's dilemma, which helps understand the conflict and cooperation between the peers of BitTorrent.

2.1 Fragment Exchange in BitTorrent

In BitTorrent, the peers that download the same file form

---

[*] This paper was supported in part by NSF grant ITR-0121643.

SIGCOMM'05 Workshops, August 22–26, 2005, Philadelphia, PA, USA.

	Cooperate	Defect
Cooperate	R=3, R=3	S=0, T=5
Defect	T=5, S=0	P=1, P=1

Table 1: Payoff matrix for the prisoner's dilemma. In each cell, the first entry separated by a comma corresponds to the row while the second entry to the column.

a distribution mesh in order to exchange the *fragments* into which the file is segmented. As peers have different sets of fragments, they exchange what they have for what they need. Thus, downloading a file means exchanging the fragments until the whole file is reconstructed. A node that has the whole file and only uploads is referred to as a *seed*.

To join the distribution mesh of a target file, a peer takes the following steps. First, it downloads the `.torrent` file of the target file from a well-known web site or by any means. This `.torrent` file, much smaller than the target file, contains the meta-data such as the *tracker* address and the cryptographic hashes that verify the integrity of fragments. Second, it connects to the tracker to retrieve a list of peers that have already joined the distribution mesh. Third, it connects to the peers on the list to become part of the distribution mesh. Neighbors notify each other of which fragments have been downloaded so that they can decide what fragments to exchange. Thereafter, it begins the download, exchanging fragments with its neighbors.

2.2 Prisoner's Dilemma

We illustrate the prisoner's dilemma with a two-person trade example adapted from Hofstadter [9]. Suppose two traders intend to exchange their items. As the exchange occurs through a magic channel, one must decide whether to send or not without knowing the decision of the other. In other words, the exchange is atomic. If they both send their items, they benefit from the exchanged items. If their decisions differ, however, one who decides not to send earns both items while the other ends up with nothing. If they both decide not to send, no one loses anything, but the desired exchange never occurs. This situation is generalized as a payoff matrix in Table 1, in which "cooperate" corresponds to the decision of sending an item while "defect" to that of not sending. The payoff assignment (R for reward, T for temptation to defect, S for sucker's payoff, and P for punishment for mutual defection) reflects the utility values earned after the exchange. Note that one always benefits more by defecting than cooperating regardless of how the other chooses to behave because $T > R$ and $P > S$. Thus, any rational (reasoning, self-interest) trader should choose to defect. This situation is a dilemma in that two rational traders are destined to earn P's instead of R's, which could be earned if they both cooperated.

The apparent impossibility of cooperation among rational traders may be turned around if the exchange is repeated because traders are now concerned with the future. The repeated exchange is generally referred to as an iterated prisoner's dilemma (IPD). To test if cooperation emerges in IPD, Axelrod [3, 2] ran computer tournaments in which every pair in a pool of players is subjected to repeated exchanges the same number of times. It is assumed that every player can recognize other players and recollect the past experience with them. That is, a player knows whether the opponent in the current exchange cooperated or defected in the previous exchanges. For every exchange, players accumulate payoffs to which the pair of their decisions ("cooperate" or "defect") corresponds in the payoff matrix of Table 1. At the end of all exchanges, the player who have accumulated the most payoffs is declared the winner.

To diversify the population, Axelrod solicited submissions, in form of computer programs, from game theorists and the public. Two round-robin tournaments were conducted with 14 entries and 62 entries, respectively, which varied in length and complexity. In both tournaments, the winning entry was TIT-FOR-TAT, which was one of the simplest, acting as follows:

> *In the first exchange, it always cooperates. Thereafter, it does what the other player did in the previous move.*

This surprisingly simple strategy has consistently outperformed other players of various type. An interesting observation is that TIT-FOR-TAT cannot defeat any single opponent because it always cooperates first and continues to do so until the opponent defects. Nevertheless, it is able to win the tournaments. At the core of being a winning strategy is not so much defeating opponents one by one as cultivating cooperation with them.

2.3 BitTorrent as Iterated Prisoner's Dilemma

To formalize the similarity between the IPD tournaments and the download process in BitTorrent, we relate them by building a payoff matrix for the fragment exchange in BitTorrent. Let d (>0) denote the utility of downloading a fragment and u (>0) denote the negative utility that represents the cost incurred in uploading a fragment. The components of the payoff matrix are assigned as follows:

$$R = d - u, \quad T = d, \quad S = -u, \quad P = 0.$$

In general, if a payoff matrix is to capture an IPD, it must satisfy two conditions. The first condition, $T > R > P > S$, is required to be a PD because, for example, a player would rather cooperate than defect if $T \leq R$. The rest of the inequality chain can be similarly justified. The second condition, $2R > S + T$, is required to be an IPD because otherwise, two players would earn more payoffs by alternating between cooperating and defecting than by always cooperating.

The payoff matrix we present satisfies the two conditions above, provided $d > u$, which should be true of those who join the BT network. After all, the peer-to-peer computing, including BT, has emerged as users can share their resources in return for some benefit (i.e., $d > u$).

3. INCENTIVE MECHANISM OF BITTORRENT

As they are all interested in the same file, peers rely on each other more directly in BT than in search networks such as Gnutella. Thus, the incentive mechanism is more appropriate to implement. In this section, we first explain the incentive mechanism that is implemented in the original BT code and then propose a new mechanism that is inspired by TIT-FOR-TAT, the winner of the IPD tournaments.

3.1 Original Mechanism

The building block of the incentive mechanism in BitTorrent is *choking* (not uploading) and *unchoking* (uploading). A peer maintains the current download rates from all its links. Based on this information, it unchokes the b links with the highest download rates (b defaults to 7 or smaller). All the other links are choked except for one that is allowed by a mechanism called the *optimistic unchoking*, the purpose of which is to find a "better" link. The period of the optimistic unchoking should be sufficiently long (30 seconds in BitTorrent 4.0.0) so that this link may be put on the unchoking list of the other peer. If it downloads from this link at a higher rate than some of the b links, this new link replaces the link with the b-th highest rate. Otherwise, another link is chosen for the optimistic unchoking in a round-robin fashion. Cohen [6] explains more details about the incentive mechanism of BT.

3.2 Proposed Mechanism

We propose an incentive mechanism that is as simple as the winning entry TIT-FOR-TAT of the IPD tournaments. In our mechanism, peers maintain the upload amount u and the download amount d for each link. We define the *deficit* of a link as $u-d$. If the constant c denotes the size of a fragment, a peer ensures that the deficit of every link is restricted up to a certain bound at any time:

$$u - d \leq f \cdot c$$

where f (≥ 1) is called a *nice factor*. Within this condition and the maximum upload rate allowed, the peer uploads evenly to all links as much as it can. This factor determines the amount that a peer is willing to risk for a chance to establish cooperation. Although neighbors may be tempted to take advantage of this "nice" peer, they will benefit more through the repeated exchange of fragments if they cooperate.

3.3 Comparison

Since the effectiveness of a strategy is relative, singling out one best strategy is difficult. Nevertheless, we may be able to find some properties that are shared by good strategies. Once such properties are identified, they can be used as guidelines to design and understand strategies.

As Axelrod identifies [2], the high-scoring entries in the IPD tournaments have in common the following four properties.

- They are *nice*, which means that they cooperate as long as the opponents do. In other words, they never defect first. This property not only brings individual success but also enables cooperation to emerge.

- They are *retaliatory*, which means that they stop cooperating if the opponents defect. This property prevents them from being exploited.

- They are *forgiving*, which means that they cooperate again if the opponents resume to cooperate after mutual defection. This property helps reconstruct trust.

- Their behavior is *clear*, which means that they signal to the opponents that they act reciprocally. This property leads the opponents, particularly those who "probe" others, into cooperation.

While our proposed mechanism has the four properties, the original mechanism lacks them because of its indirect reciprocity in choking and unchoking. In the original mechanism, upon finding a better link, it replaces an existing link, which indicates that the peer defects first (i.e., not nice). This replacement results in mutual defection. Reconstructing cooperation may be difficult as it requires optimistic unchoking from one peer and the considerable upload from the other within the 30-second window of opportunity. On the other hand, the optimistic unchoking keeps the original mechanism from being retaliatory. A 30-second unconditional upload can be a significant resource leak if it is not compensated for. Moreover, since the links are chosen in a round-robin fashion, free riders are given repeated benefit. That is, the amount of upload is not restricted.

The next section complements this qualitative analysis with the experimental results from running the actual code on PlanetLab.

4. EVALUATION

We evaluate the incentive mechanisms in a game-like environment. A game begins with all peers downloading at the same time and ends when the last peer completes the download. After a game, the performance of the peers is judged by the two metrics: the download completion time and the upload amount. Minimizing both the metrics simultaneously should be difficult under an effective incentive mechanism. In other words, the two metrics should be inversely proportional. Throughout this section, we refer to the original incentive mechanism in Section 3.1 as OLD and our proposed one in Section 3.2 as NEW.

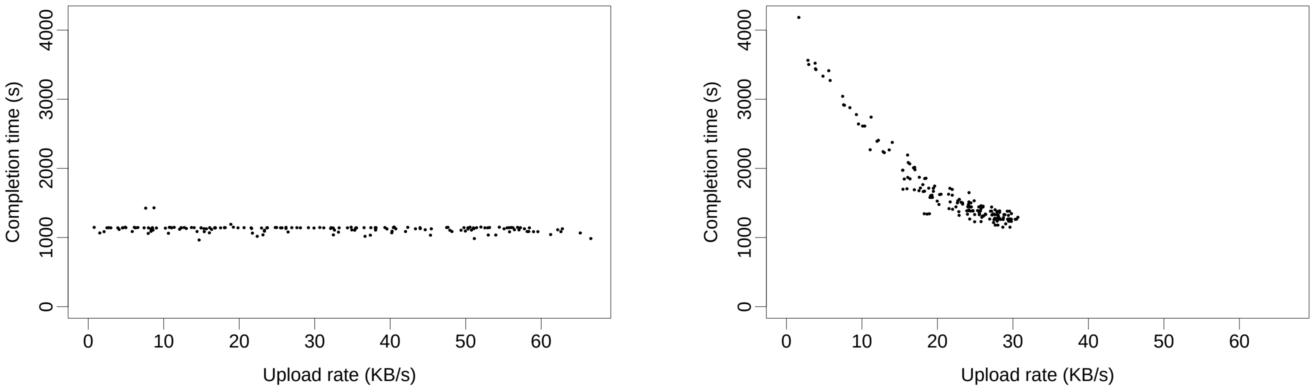
4.1 Assumptions

In this evaluation, we assume that peers do not exploit the game in the following ways, some of which merit further investigation. First, one user injects multiple peers and controls them to its advantage [7], or in general, peers collude in any way. Second, peers assign `max_initiate` a large value and `rerequest_interval` a small value to obtain more neighbors than normally assigned. Third, peers modify or fabricate the file fragments illicitly in order to hamper the progress of others or boost its own progress [19]. Last, peers vary their strategy according to the download progress. For example, they may defect at or near the end of downloading. In fact, the game theory indicates that when rational players are engaged in the fixed round game, the defection cascades up to the first round and cooperation never occurs [12].

4.2 Experimental Setup

A game consists of one tracker, one seed, and about 170 downloaders. After the tracker and the seed launch, the downloaders start to download by contacting the tracker at the same time. The tracker and the seed run throughout the game while the downloaders exit immediately after they complete the download. We use PlanetLab nodes to host the downloaders, one per node [15]. The target file is the 33 MB `linux-2.6.8.1.tar.bz2`, divided into 128 KB fragments.

The tracker, the seed, and the OLD downloaders run the BitTorrent 4.0.0 code. The code remains unchanged except for the data collection and the automatic exit after the download completion. The NEW downloaders run the modified code in which the choking is disabled while the *nice* invariant is implemented as explained in Section 3.2



(a) A game with 165 OLD downloaders

(b) A game with 171 NEW downloaders

Figure 1: Correlation between completion time and upload rate in pure population games. Each point represents a downloader.

TYPE	PARAMETER	VALUE
Tracker	`max_give`	200
Seed	`max_upload_rate`	100
	`max_uploads`	200
	`max_initiate`	200
	`max_allow_in`	200
	`min_peers`	200
	`rerequest_interval`	30
Downloader	`max_initiate`	20
	`max_allow_in`	40

Table 2: Parameters and their values

($f = 1.0$).

All the parameters are set to the default values except for the ones listed in Table 2. For the tracker and the seed, the changed parameter values, along with the extension to the code, enable the seed to contribute evenly to every downloader at any moment. That is, in a mesh composed of D downloaders, the seed uploads to each downloader at the rate of $\frac{100}{D}$ KB/s. The parameters for the downloaders limit the number of neighbors up to 40 (the default is 80), as the mesh is relatively small.

According to the population mix, the game is of two types. First, in the *pure population game*, the population is either entirely OLD or entirely NEW. Peers independently choose the maximum upload rate uniformly randomly from the range $[1, 100]$ in the beginning of the game. Second, in the *mixed population game*, the population is a mixture of three classes: OLD, NEW, and FRD. The FRD (free rider) downloaders run the same code as OLD—only with the maximum upload rate set to 4 KB/s. For OLD and NEW, the maximum upload rate is set to 100 KB/s.

4.3 Results

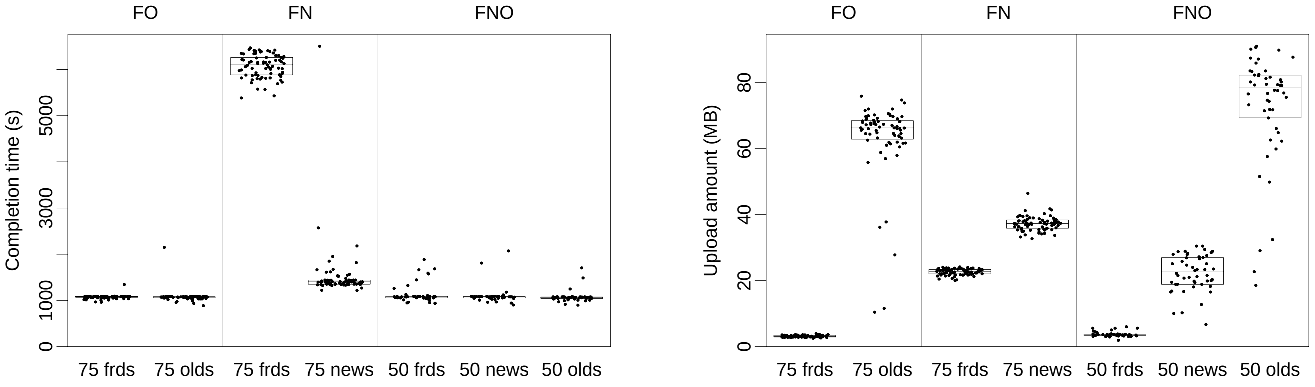
The results of the two **pure population games** are shown in Figure 1, focusing on the correlation between the completion time (the duration of a peer's download) and the upload rate (the total upload amount divided by the completion time). Each point in the plots corresponds to one downloader. Figure 1(a), in which all downloaders belong to the OLD class, shows that all downloaders finish about the same time regardless of the upload rate. As the contribution is not rewarded, nor the lack thereof is punished, the OLD downloaders have no incentive to upload. In contrast, Figure 1(b) clearly shows that downloaders with the higher upload rate finish earlier.

Although the seed uploads at 100 KB/s and the maximum upload rates for the downloaders are distributed uniformly over $[1, 100]$, the actual upload rate is distributed over the range $[0.78, 66.56]$ in OLD and $[1.62, 30.65]$ in NEW because the "effective" upload rate from the seed is lower than 100 KB/s due to the inefficient fragment scheduling [8]. The NEW downloaders suffer more than the OLD downloaders from the inefficient scheduling because the lack of diversity in fragments hurt the tit-for-tat strategy more severely. We believe that more efficient scheduling will narrow the gap of the upload rate distributions between OLD and NEW.

The average completion time is shorter in the former game (1123 vs 1672 seconds for means and 1139 vs 1441 for medians) largely because of the skewed distribution of upload rate in the NEW downloaders. From this result, one may argue that the original incentive mechanism represented by the OLD downloaders is better. The problem is, however, that the system comprised of OLD downloaders is not sustainable in that as they learn to be free riders, the whole system will eventually suffer "the tragedy of the commons." In contrast, the strong incentive to avoid free riding in NEW will keep the system sustainable. Therefore, we argue that NEW is better than OLD in this evolutionary aspect.

Drawing analogies between downloading and earning and between uploading and spending, we define the *deficit* of a node as the total download amount subtracted from the total upload amount. The amount received from the seed does not account for the deficit. The node deficit shows the degree of free riding. A node with a large absolute value of negative deficit can be considered a free rider. The number of downloaders with -10 MB deficit or less is 61 in OLD vs 0 in NEW. The minimum deficit is -32 MB in OLD (i.e., nearly no upload) vs -9 MB in NEW.

The results of the **mixed population games** reinforce



(a) Download completion time

(b) Upload amount

Figure 2: Results of three mixed population games (FO, FN, and FNO). Each game consists of 150 downloaders with a different mixture of population. Points are given horizontal jitter to show their volume more clearly.

the susceptibility of OLD and the robustness of NEW. Figure 2 shows the results of three mixed population games, labeled FO (a game of FRD vs OLD), FN (FRD vs NEW), and FNO (FRD vs NEW vs OLD), respectively. While each point represents one downloader, each superimposed box indicates the distribution of points with the bottom line placed at 25% quantile, the middle line at 50%, and the top line at 75%. Thus, half the points in the middle are captured inside the box.

In the FO game, in which 150 downloaders are halved into two classes FRD and OLD, the FRD and the OLD downloaders have almost the same average download completion time (1072 vs 1075 seconds) whereas the FRD downloaders upload much less on average than OLD (3 MB vs 64 MB). As Figure 2(b) shows, some OLD downloaders upload less than others in the same class because they have the limited upload capacity less than `max_upload_rate` (100 KB/s).

By contrast, in the FN game, in which 150 downloaders are halved into two classes FRD and NEW, the FRD downloaders finish much later than the NEW downloaders (the average completion time is 6068 vs 1516 seconds). One NEW downloader finishes very late at 6502 seconds because as the only NEW downloader surrounded by FRD downloaders it must play like a free rider. This situation would rarely happen in an open system in which new downloaders keep coming to join in. With this downloader excluded, the average completion time of the NEW downloaders decreases to 1449 seconds, which is still higher than that of the OLD downloaders in the FO game because the OLD downloaders tend to maximize the utilization of their upload capacity regardless of the reciprocal action by their neighbors. Certainly, such tendency makes the system more efficient for the time being, but we believe that it will not be sustainable because it is bound to induce free riding. As for upload amount, although the FRD downloaders upload less than the NEW downloaders (23 MB vs 37 MB), the gap between the two numbers narrows compared with the gap in the FO game (from 61 MB down to 14 MB). As the free riders run longer, they eventually upload considerable amount despite the low maximum upload rate.

The FNO game suggests that OLD downloaders might prefer NEW as it saves the upload bandwidth. If enough downloaders migrate to NEW, free riders will start to suffer as shown in the FN game. Then, the system gradually becomes like the FN game that is more resistant to free riders and hence sustainable.

5. RELATED WORK

The concept of prisoner's dilemma, since developed in 1950 [11], has applied to various disciplines such as economics, biology, psychology, and politics as it captures some fundamental conflict of interests. In this paper, we apply it to BitTorrent, which opens up the same kind of dilemma between the eagerness to download and the unwillingness to upload.

Cohen [6], who developed BT, explains its incentive mechanism, which we evaluate in comparison with our proposed mechanism. Although others [20, 4] also propose similar mechanisms bounding the difference between upload and download amounts, we provide an insight into such mechanisms using the iterated prisoner's dilemma as well as direct comparison with the original mechanism using an actual implementation. Shneidman et al. discuss the concept of faithfulness, making a case for BT [19]. While they examine BT for unfaithful components, we show that the current mechanism of BT as a whole may be unfaithful. Nielson et al. imply the relationship between BT and IPD in the context of the service maturation taxonomy [13]. In this paper, we experimentally evaluate the effectiveness of a strategy based on IPD.

Other aspects of BT have also been studied. Qiu and Srikant study the scalability and performance of BT using a simple fluid model [17]. Gkantsidis and Rodriguez use network coding to improve the fragment scheduling [8]. The back-off strategy of Slurpie [18] may make BT more scalable while it requires peers to be cooperative. Izal et al. [10] and Pouwelse et al. [16] provide empirical studies.

6. CONCLUSIONS AND FUTURE WORK

By relating BitTorrent to the iterated prisoner's dilemma, we provide a game-theoretic framework for understanding the incentive mechanism of BitTorrent. Under this framework, we present the properties of good strategies and, based on them, compare the two incentive mechanisms. Furthermore, the experimental results show that the original incentive mechanism is susceptible to free riding whereas our proposed mechanism is more robust against it. While we believe that we provide the evidence, we are further investigating how to model and understand the interactions between peers and the long-term evolution of them in a rigorous, comprehensive manner.

7. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments that have improved the paper.

8. REFERENCES

[1] Eytan Adar and Bernardo A. Huberman. Free riding on Gnutella. *First Monday*, 5(10), October 2000.

[2] Robert Axelrod. *The Evolution of Cooperation.* Basic Books, 1984.

[3] Robert Axelrod and William D. Hamilton. The evolution of cooperation. *Science*, 211:1390–1396, 1981.

[4] Ashwin R. Bharambe, Cormac Herley, and Venkata N. Padmanabhan. Analyzing and improving BitTorrent performance. Technical Report MSR-TR-2005-03, Microsoft Research, February 2005.

[5] BitTorrent. `http://bittorrent.com`.

[6] Bram Cohen. Incentives build robustness in BitTorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[7] John R. Douceur. The Sybil attack. In *Proc. the 1st International Workshop on Peer-to-Peer Systems*, 2002.

[8] Christos Gkantsidis and Pablo Rodriguez Rodriguez. Network coding for large scale content distribution. In *Proceedings of IEEE Infocom*, Miami, FL, March 2005.

[9] Douglas R. Hofstadter. The prisoner's dilemma computer tournaments and the evolution of cooperation. *Scientific American*, 248(5):14–20, May 1983.

[10] M. Izal, G. Urvoy-Keller, E.W. Biersack, P.A. Felber, A. Al Hamra, and L. Garcés-Erice. Dissecting bittorrent: Five months in a torrent's lifetime. In *Proceedings of the 5th Passive and Active Measurement Workshop*, April 2004.

[11] Steven Kuhn. Prisoner's dilemma. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Fall 2003.

[12] Roger B. Myerson. *Game Theory: Analysis of Conflict.* Harvard University Press, September 1991.

[13] Seth James Nielson, Scott A. Crosby, and Dan S. Wallach. A taxonomy of rational attacks. In *Proceedings of the 4th International Workshop on Peer-To-Peer Systems*, February 2005.

[14] Adam Pasick. File-sharing network thrives beneath the radar. `http://in.tech.yahoo.com/041103/137/2ho4i.html`, November 2004. LONDON (Reuters).

[15] PlanetLab. `http://www.planet-lab.org`.

[16] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips. The BitTorrent p2p file-sharing system: Measurements and analysis. In *Proceedings of the 4th International Workshop on Peer-To-Peer Systems*, February 2005.

[17] Dongyu Qiu and R. Srikant. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 367–378, New York, NY, USA, 2004. ACM Press.

[18] Rob Sherwood, Ryan Braud, and Bobby Bhattacharjee. Slurpie: A cooperative bulk data transfer protocol. In *Proceedings of IEEE Infocom*, 2004.

[19] Jeffrey Shneidman, David C. Parkes, and Laurent Massoulié. Faithfulness in internet algorithms. In *PINS '04: Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 220–227. ACM Press, 2004.

[20] Karthik Tamilmani, Vinay Pai, and Alexander Mohr. SWIFT: A system with incentives for trading. In *Proceedings of the 2nd Workshop on Economics of Peer-to-Peer Systems*, June 2004.