SWIFT: A System With Incentives For Trading

Karthik Tamilmani, Vinay Pai and Alexander Mohr
Department of Computer Science
Stony Brook University
{tamilman, vinay, amohr}@cs.stonybrook.edu

Abstract

In this paper, we present the design of a credit-based trading mechanism for peer-to-peer file sharing networks. We divide files into verifiable pieces; every peer interested in a file requests these pieces individually from the peers it is connected to. Our goal is to build a mechanism that supports fair large scale distribution in which downloads are fast, with low startup latency. We build a trading model in which peers use a pairwise currency to reconcile trading differences with each other and examine various trading strategies that peers can adopt. We show through analysis and simulation that peers who contribute to the network and take risks receive the most benefit in return. Our simulations demonstrate that peers who set high upload rates receive high download rates in return, but free-riders download very slowly compared to peers who upload. Finally, we propose a default trading strategy that is good for both the network as a whole and the peer employing it: deviating from that strategy yields little or no advantage for the peer.

1 Introduction

In recent years, peer-to-peer file sharing networks like Kazaa, Napster, and Gnutella have become very popular networks for exchanging files on the Internet. These networks are distributed and allow users to download copies of files from other users who share these files. Users want to obtain their desired files as quickly as possible; however, their download rate is constrained by the upload rates of other peers in the system.

Some peers use peer-to-peer networks for their own benefit without offering any services in return and are called free-riders. Free-riding exists due to the lack of concrete incentives to contribute. In the past, economists have analyzed the free riding problem that exists in society [3, 10, 12, 16]. Recently, networking researchers have been measuring free-riding in peer-to-peer networks. Saroiu *et al.* [18] estimated that 20–40% of Napster users and up to 70% of Gnutella users shared little or no content. Huberman and Adar [1] found that nearly 50% of responses are returned by 1% of the sharing hosts and that nearly 98% of the responses were returned by 25% of the sharing hosts.

Many economic frameworks [5, 11], especially currency systems [21] and incentive models [6, 14, 15, 17], have been proposed to solve the free riding problem. Some of these incentive models describe rational behavior in which peers employ game theory within the limits of the protocol and try to maximize their download speeds. Buragohain *et al*. [4] have shown that systems that employ differential incentives will eventually operate at a good Nash Equilibrium.

Much work has focused on rational behavior, but there is at least one more type of behavior. *Obedient peers* [20] tend to use the default settings in client software and make no attempt to be “rational.” As system designers, it is important to recognize that our goal is not merely to achieve a Nash equilibrium when all peers are rational; rather, we want a system in which obedient peers operate near the point of self-interested peers, incentives to defect. such that deviating from the default behavior gives them little or no advantage.

However, we also want a system that scales well as the number of peers increases. Ideally, a source could upload a file once and it would propagate to millions of peers, with each peer contributing equally. More realistically, each peer should contribute comparably. We call this fair large scale distribution.

The goals of SWIFT are five fold. First, it should support fair large scale distribution of files. Second, download rates should be as fast as possible. Third, new peers joining the system should not have to wait long before they can begin downloading. Fourth, it should be robust to attacks by malicious users. Finally, it should have a default trading strategy that is good for both the network as a whole and the peer employing it: deviating from that strategy should yield little or no advantage for the peer.

In this paper, we present the design of SWIFT and show how it achieves those goals. In Section 2, we describe our file trading model and the assumptions we make. In Section 3, we highlight some strategies that peers may adopt and in Section 4 show that peers should take risks periodically as part of the default strategy. In Section 5, we describe a simulation of SWIFT and report our experimental results. In Section 6, we discuss how our system compares to related work. In Section 7 we mention future work and finally, in Section 8, we conclude.

2 The File Trading Model

Some file sharing networks such as Napster and Gnutella allow peers to download whole files from a single peer only. More recent networks such as BitTorrent and Kazaa allow peers to grab pieces of the file from multiple sources. Pieces provide a finer granularity for transactions between peers, which especially suits our trading model. In our system, we assume that a file is broken into pieces of equal size and that the authenticity of each piece can be verified by a scheme such as a cryptographic hash or Merkle tree [13]. Our model is pull-based in that peers advertise the pieces that they have; other peers then request specific pieces from them. We further assume that the file sharing network has some mechanism in place for peers to discover fellow peers and join the system.

Some peers only contribute to the system and have no desire to download from the network. For example, many Linux distributions are available as ISO files from the organizations that produce them—their servers act as distributors. In the rest of this paper, we only consider strategies for peers who are interested in downloading files from the network.

In SWIFT, we denote peers who exchange pieces as *traders*. A trader's objective is to obtain copies of files he is interested in by exchanging pieces he has for pieces he wants. Rather than negotiate a piece-for-piece trade as in a barter system, we assume each trader maintains a credit (a pairwise currency) for every peer to which it is connected. When the host receives and verifies a piece from a peer, the host increases the credit rating of that peer in proportion to the size of the received piece. Similarly, when a host fulfills a remote peer's request, the host decreases the credit rating of the peer by the size of that piece. A remote peer's request is satisfied only if it has accumulated credit greater than or equal to the requested piece's size. The pairwise currency is currently used only to reconcile current trading imbalances and not for long term savings, although it could be extended across multiple sessions to trade different files, but this is not critical to the working of our system.

For example, it was observed in the BitTorrent [6] network that for the first three days after the release of the RedHat 9.0 ISO, there were always more than 2500 peers simultaneously downloading that 1.6GB file, with a peak of 4400 peers [8]. A similar example occurs when Microsoft releases a large service pack or update, after which millions of computers running the Windows operating system will all be interested in obtaining that update in a short time frame. As evidenced by the spread of the recent Witty worm [19] less than 24 hours after a patch was produced for the vulnerability it exploited, these machines should obtain patches as quickly as possible.

In the next section, we introduce three different trading strategies and discuss which to choose.

3 Trading Strategies

We parameterize the behavior of peers based on how they extend credit to their neighbors. For every byte a peer receives, it extends the sender α bytes of credit in return. We call α the *repayment ratio*. In addition, it expends a fraction β of its total upload capacity U_{max} on *largesse* by uniformly distributing free credit to all N of its neighbors. Finally, a peer also extends every neighbor γ bytes of one-time credit the first time they interact.

Peer Behavior	α	β	γ
Free Rider	0	0	0
Paranoid Trader	1	0	0
One-time Risk-taking Trader	1	0	1
Periodic Risk-taking Trader	1	$0 < \beta \leq 1$	1
Distributor	N/A	1	1

Table 1: This table summarizes the values of the parameters for some common peer behaviors.

The maximum number of bytes $u_{AB}(t)$ that peer A is willing upload to its neighbor B at time t, having received $d_{AB}(t)$ from B, is given by the equation:

$$u_{AB}(t) = \alpha d_{AB}(t) + \frac{\beta U_{max}}{N_A} t + \gamma. \tag{1}$$

Note that time t here is meant to represent wall-clock time and not a tit-for-tat mechanism in which time is divided into rounds.

Free-riders, who do not upload, have repayment ratio α, largesse rate β, and one-time free credit γ of zero. Distributors, who have no interest in downloading, have $\beta = 1$ and spend all their upload bandwidth on distributing pieces to their neighbors. Traders who are mainly motivated by their desire to download a file as quickly as possible lie between these two extremes. Based on their choice of parameters, we classify them as paranoid traders, one-time risk-taking traders, or periodic risk-taking traders.

Paranoid traders

Paranoid traders are reciprocative players that wait until they receive a valid piece from a peer before offering to send an equal amount back. They have repayment ratio $\alpha = 1$ and never give out free credit ($\beta = 0$, $\gamma = 0$). This conservative strategy ensures that they will never upload more to a peer than they receive from it and thus will never be taken advantage of.

One-time risk-takers

Another strategy is for a peer to extend one piece of free credit to a peer the first time it is encountered to encourage them to trade. However, there is a chance that the peer will never receive a piece in return, so we call these traders one-time risk-takers. They set α and γ to 1, and β to 0.

Periodic risk-takers

Finally, some traders may be willing to give out free pieces periodically. These traders dedicate a fraction $\beta > 0$ of their upload bandwidth giving out free pieces to their neighbors. We call this type of free credit *largesse*.

The choice of a strategy

Table 1 summarizes the values of α, β and γ for the different types of peers that we have described. It is clear that if the system consists solely of paranoid traders, everyone will wait for their neighbors to make the first move and the system will be deadlocked.

At first glance, it would appear that one-time risk taking is sufficient to break the deadlock by giving peers a basis to start trading. However, we show through simulation in Section 5 that one-time risk-taking does not completely eliminate the deadlock. A peer that is not connected to a distributor will receive free pieces from its neighbors when it first joins. However, it is possible that it will acquire pieces that none of its other neighbors are interested in, and will then be unable to trade and make further progress.

The one-shot free credit is also insufficient because of the peer identification problem. Allowing a peer to choose its identity will make the system susceptible to a Sybil attack [7]. One way to alleviate this problem is to use the IP address of a peer as its identifier. However, when there are many peers behind a Network Address Translator (NAT), all of them use the same IP address, so only one would receive the one-time free credit and bootstrap into the file sharing network, leaving the others to starve. On the other hand, a periodic risk-taker could distribute the IP's share of the largesse equally to each instance behind a NAT so that all of them are able to join the system.

Finally, if transport across the network is unreliable or subject to corruption, perfect accounting is not guaranteed. For instance, a peer may upload a piece and bill its neighbor for it, but the piece fails the cryptographic checksum and the peer receives no credit for it. The peer may then be stranded with no pieces to trade and no credit with any of its peers, resulting in starvation.

Adopting the periodic risk-taking strategy increases the possibility of wasting upload bandwidth on free riders, but we show in the next two sections, via mathematical analysis and simulation, that the advantages of this variant of Tit-for-tat [2] outweigh this potential drawback while maintaining robustness against a wide range of competing strategies.

4 Analysis

Let us now consider a homogeneous file trading system of N peers with upload and download capacities of $U_{max} = D_{max}$. Given a default strategy of periodic risk-taking, we analyze how that strategy interacts with others.

4.1 Bounds on Incentives to Defect

The bounds on the incentive for peers who wish to maximize their download rates to defect from the periodic risk-taking strategy can be made arbitrarily small. Consider the case of a mixed network of rational peers and periodic risk-takers. Let σ be the fraction of periodic risk-takers in the system, each of which contributes a fraction β of their upload bandwidth as largesse. If the share ratio of a peer is defined as the ratio of bytes uploaded to bytes downloaded, then the share ratio of periodic risk-takers $r_{periodic}$ is given by

$$r_{periodic} = \frac{1}{(1-\beta)+\beta\sigma}$$

and the share ratio of rational traders $r_{rational}$ by

$$r_{rational} = 1 - \beta\sigma.$$

When $\beta = 0.1$, then in the extreme case of one rational peer among many periodic risk-takers, the greedy trader's share ratio is approximately 1.1. Similarly, in the other extreme of one periodic risk-taker among legions of rational peers, the risk-taker's share ratio is approximately 0.9. Clearly, these bounds can be made arbitrarily close to one by decreasing β.

4.2 Paranoid Traders vs. Periodic Risk-takers

To show that a weak Nash equilibrium [9] can exist between paranoid traders and periodic risk-takers, we assume that each peer uses fair queuing among its neighbors to share its upload bandwidth C.

We observe that paranoid traders will only trade with periodic risk-takers, as two paranoid traders will never risk a piece on each other. Thus, a paranoid trader will trade with σN periodic risk-takers, while a periodic risk-taker will trade with $N-1$ peers.

If we assume the largesse rate β is sufficiently small, then each connection's capacity will be limited by the fair rate of $\frac{C}{N-1}$ that periodic risk-takers assign to each connection. Periodic risk-takers then achieve upload and download rates of $(N-1)(\frac{C}{N-1}) = C$, whereas paranoid traders achieve rates of $(\sigma N)(\frac{C}{N-1}) = \sigma C$ (as $N \to \infty$). Taking β into account and assuming the worst-case scenario in which none of the largesse is repaid, the download rate of periodic risk-takers falls to $(1-\beta)C$. Paranoid traders will download more quickly than periodic risk-takers when $\sigma > 1-\beta$ and download less quickly when $\sigma < 1-\beta$, so the system attains a weak Nash equilibrium point with respect to download speeds when $\sigma = 1-\beta$. For small β, the equilibrium point is a network consisting almost entirely of periodic risk-takers.

4.3 Incentives Not to Free-ride

Consider now a system consisting of fraction σ periodic risk-takers and fraction $1-\sigma$ free-riders. Each free-rider will be able to download at a rate of $\frac{\beta C}{N-1}$ from each of the σN risk-takers, which results in a total download rate for the free-rider of $\beta\sigma C$ (as $N \to \infty$). Although free-riders can achieve share ratios of zero, they will download at a rate much lower than the risk-takers. For example, if $\sigma = 0.5$ and $\beta = 0.1$, they will download at a rate only 5% that of the risk-takers. Furthermore, as the number of free-riders increase, the incentive to become a risk-taker increases!

5 Simulation Results

We built a discrete-time simulator for our system. The simulator distributes bandwidth evenly between all connections and assumes that the bottleneck is always at the end-hosts' connection to their ISP. Download capacity, upload capacity, repayment ratio α, largesse rate β, and one-time free credit γ can be set on a per-link basis. In practice, we used the same values of α and β for all links originating from a given node, while using a random value of γ to avoid synchronization artifacts when all peers accumulate enough largesse to download a piece simultaneously. We ran all of our experiments

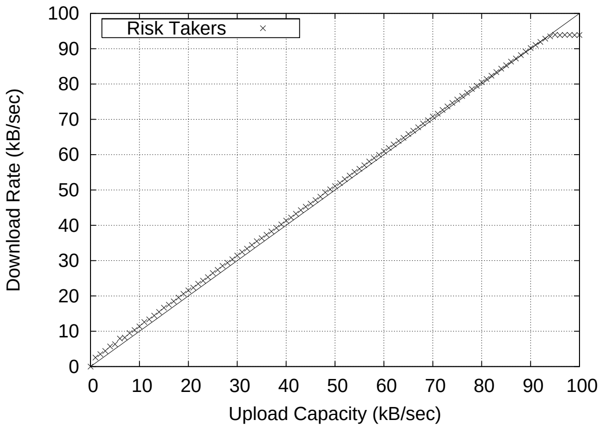

Figure 1: The download rate obtained by a peer as function of its upload capacity. Every "x" represents one peer. The diagonal line represents $x = y$.

with a single seed and 100 peers who want to download the file. In each experiment we report average rates after the system has achieved a steady state.

5.1 Download vs. Upload Rates

In our first experiment we show that peers have a strong incentive to upload as much as they can. All 100 peers used repayment ratio $\alpha = 1$, largesse rate $\beta = 0.01$, and a random one-time free credit γ between 1 and 2. All peers had download capacities of 100 kB/sec, but upload capacities were uniformly limited to values between 1 and 100 kB/sec. The topology used was a complete graph and the file had 100,000 pieces.

Figure 1 shows the resulting download rates obtained by peers as a function of their upload capacity, with the straight line representing equal upload and download rates. It is evident that periodic risk-takers with upload capacities less than 94 kB/sec receive download rates comparable to their upload capacity, with most peers receiving slightly more than they upload because of the free pieces they receive from the seed. In SWIFT, peers clearly have incentives to set high upload rates.

Peers with upload capacity greater than 94 kB/sec operate below slightly capacity. Our preliminary analysis indicates that this degradation is likely an artifact of the random piece picking strategy that we employed in our simulator. We suspect that the problem would be alleviated if the piece picking algorithm were to take into account the frequency of pieces in the system, with a bias towards rarer ones.

5.2 Paranoid Traders vs. Periodic Risk-takers

In Section 4.2, we claimed that, in a mixed network of paranoid traders and periodic risk-takers, the risk-takers download faster. We modified the previous experiment to demonstrate that claim by changing half the peers into paranoid traders who did not upload a piece unless they first received one.

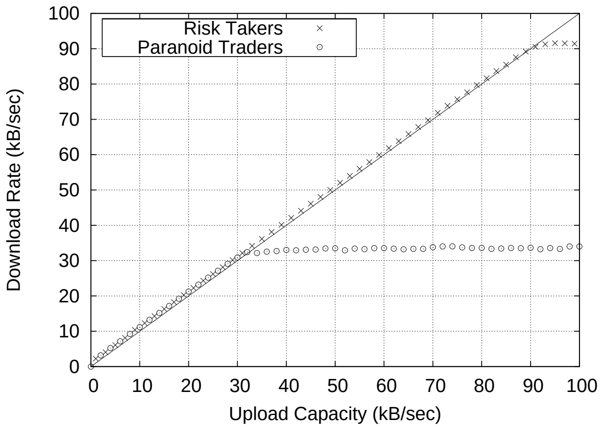

Figure 2: The download rate obtained by a peer by paranoid traders and periodic risk-takers as a function of their upload capacity. The diagonal line represents $x = y$.

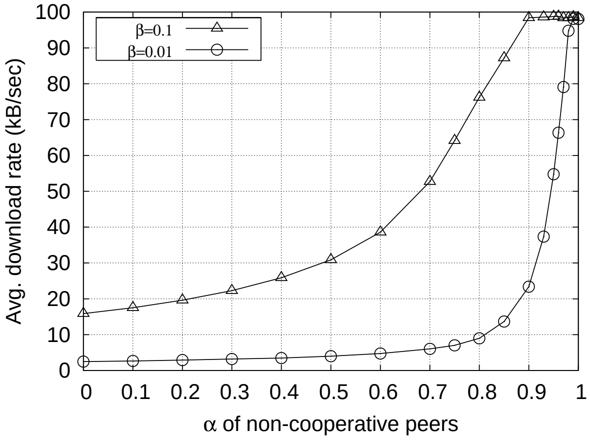

Figure 3: The average download rate received by non-cooperating peers as a function of their α.

Figure 2 shows the resulting download rates obtained by peers as a function of their upload capacity. The average download rate of paranoid traders was 28.5 kB/sec, whereas the average download rate of risk-takers was 50.5 kB/sec. We noticed that the paranoid traders traded with only with the risk-takers and thus downloaded at a much slower rate, as predicted in Section 4.2.

5.3 Effect of Non-cooperative Peers

Our third experiment studied the behavior of non-cooperative peers that use repayment ratios α other than the default value of 1. As claimed in our analysis in Section 4.1, we show that a peer has very little incentive to deviate from the default behavior. Once again, we used a complete graph. Half of the peers were obedient and used $\alpha = 1.0$ whereas the remaining half used values ranging from 0 to 0.99. In the first run all peers used a largesse rate $\beta = 0.1$ whereas in the second run they used $\beta = 0.01$. All peers had an upload and download capacity of 100 kB/sec.

As shown in Figure 3, the download rate received by non-cooperative peers was much less than 100 kB/sec for peers with small values of α, but rose sharply as α approached 1. When $\beta = 0.1$, non-cooperative peers must still upload

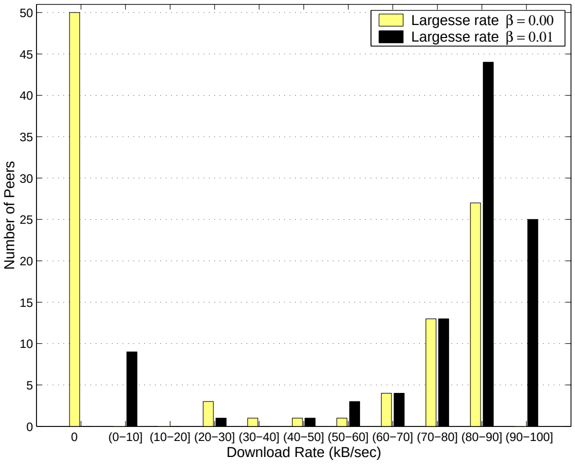

Figure 4: Distribution of peers by their download rate in two experiments. One, shown in gray, used $\beta = 0$ and the other, shown in black, used $\beta = 0.01$.

about 90% of what they receive in order to saturate their download link. With $\beta = 0.01$, the effect is more pronounced: non-cooperative peers must use a repayment ration α very close to 1 to saturate their download link. Selfish peers are quickly penalized for their non-cooperative behavior.

5.4 Incentives Not to Free-ride

In Section 4.3, we showed analytically that free-riders download at a much slower rate compared to periodic risk-takers. To demonstrate this, we ran an experiment of 100 peers with half of the peers free-riding and the other half periodic risk-takers with upload capacity of 100 kB/sec, $\alpha = 1$, $\beta = 0.1$, and γ set randomly between 1 and 2.

We observed that the free-riders downloaded at only 6 kB/sec, whereas the periodic risk-takers downloaded at 50 kB/sec. Of the 6 kB/sec that free-riders received, 1 kB/sec was received from the seed, whereas 5 kB/sec was received from periodic risk-takers as predicted in Section 4.3.

5.5 Case for non-zero β

It is quite clear that having both $\beta = 0$ and $\gamma = 0$ will deadlock the system as no peer other than a seed will ever upload a piece. However, in this experiment we now demonstrate that a simple one-time credit is not sufficient to solve this problem.

We created a random graph with one seed, 100 other peers and an average node degree of 20. All peers have upload and download capacities of 100 kB/sec.

Figure 4 shows the distribution of peers receiving various download speeds. In the absence of largesse, half the peers have download rates of zero and are deadlocked. These are peers that are not directly connected to the seed and no longer have pieces that their neighbors are interested in trading for. Since these peers never again receive a free piece, they will never reach completion. Contrast that scenario with $\beta = 0.01$ in which no peers are deadlocked and download rates are significantly faster, which shows that having all peers risk a small fraction of their bandwidth on giving away free pieces not only improves overall system performance, but is also likely to bring a high return-on-investment for the peers themselves.

6 Related Work

Many economic models have been proposed for peer-to-peer networks [4, 6, 14, 15, 17, 21]. Here we discuss a few of them.

The BitTorrent [6] network is one of the few incentive based peer-to-peer file sharing networks in widespread use on the Internet. Like SWIFT, BitTorrent also limits free riding by incorporating a tit-for-tat principle. However, the action taken when a peer uploads is different. In BitTorrent, a peer selects the peer to upload as fast as it can, but SWIFT is more fine-grained in that the peer returns only how much it owes the remote peer. In BitTorrent, if the remote peer has already saturated its upload capacity, a host continuously gives it more than necessary but gets the same download rate. In our trading system, this problem is constrained: a remote peer may download only if it has sufficient credit, which it can maintain only by uploading at the same rate.

Buragohain *et al.* propose a game theoretic framework which provides incentives through a differential service [4]. The authors model the file sharing system as an *N* player non-cooperative game in which each peer determines the utility of of a download it received based on a benefit matrix. Each b_{ij} in the benefit matrix *B* corresponds to the α in Equation 1. The authors use game theory to prove the existence of a good Nash equilibrium and prove that rational peers either converge at this equilibrium point or quit the game. However, the game theory model assumes all peers can be trusted and may not be resilient to attacks by non-cooperative or malicious peers. We take a similar approach to build the SWIFT file sharing network taking into account the non-cooperation of peers.

Another economic model due to Ngan *et al.* [14] maintains two scalar metrics corresponding to the number of objects sent and number of objects received whose difference is the debt or credit that a node has with its peer. Peers use *relationship throttling* to limit free-riding. This works well for most cases, but if a diligent peer stays on the network for long laboring with slow peers and building up credit, it may never get a chance to discover faster peers. The design relies on the altruism of intermediate nodes to download and upload blocks on behalf of other nodes. From the end user's perspective, the non-useful bandwidth expended in order to keep its debt low could be high. However, the debt-based paths constructed by these peers not only speed up transfers, but also potentially allow peers to download from others who do not directly have a debt-credit relationship.

A micropayment system called Karma [21] provides incentives based on a single system-wide scalar per peer called its karma using a micro-payment scheme. In its current form, Karma's cryptographic and accounting overheads make fine-grained transactions expensive. Our trading system could be an adjunct to such a micropayment system: in such a combined system, peers may use micropayments when trading is insufficient.

7 Future Work

We have shown in Figure 4 that a non-zero largesse rate β decreases the probability of deadlock and increases the overall system throughput. However, a high largesse rate(β) also means that a peer will waste its bandwidth by uploading to free-riders. The appropriate value of β in a real world system is not known. Although a largesse of 1% does not seem unreasonable, we will try to arrive at an optimal value.

Another important consideration is the number of traders a peer is engaged with. Ideally, a peer should be able to connect to as many peers as possible and trade with all other peers. However peers in file sharing systems usually have a finite limit on their number of simultaneous connections mainly due to control traffic overhead. To maximize its download speed a peer should ensure that its selected connections are likely to saturate its download bandwidth. The dynamics are quite different in SWIFT with the credit balance. We will study how peer connectivity changes the behavior of peers in SWIFT.

The semantics of streaming video data resembles that of file sharing networks in many ways. We will examine if a SWIFT style economic framework will work for streaming in a peer-to-peer network in a non-cooperative environment.

8 Conclusion

In this paper, we have designed a file trading system called SWIFT in which peers employ a default trading strategy that is good for both the peer employing it and the system as a whole. SWIFT accounts for peer trading deficits with credits, which results in fair large scale distribution of files. The one-shot credit that risk-taking peers provide helps new peers to rapidly take part. The periodic largesse helps peers discover faster neighbors and prevents other peers from deadlocking. Through analysis and simulation we have shown that peers who take risk and contribute to the system benefit more than those who do not. Furthermore, free riders can download only at a fraction of the download speed of peers who upload. With SWIFT, when there is insufficient upload capacity to satisfy demand, free riders are affected far more than the peers that upload and contribute to the system.

We have added SWIFT to the official BitTorrent Experimental client, version 3.2.1b, and named the result *TradeTorrent*. It is available for public download at http://mnl.cs.stonybrook.edu/project/tradetorrent/ .

Acknowledgments. The authors would like to thank Johan Pouwelse and Jeffrey Shneidman for helpful discussions and comments.

References

[1] E. Adar and B.A. Huberman. Free Riding on Gnutella. *First Monday*, 5(10), Oct 2000.

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

[3] T. C. Bergstrom. 'A fresh look at the rotten kid theorem and other household mysteries'. *Journal of Political Economy*, 1989.

[4] C. Buragohain, D. Agrawal, and S. Suri. A Game Theoretic Framework for Incentives in P2P Systems. *Proceedings of the Third IEEE International Conference on Peer-to-Peer Computing*, Sep 2003.

[5] N. Christin and J. Chuang. On the cost of participating in a peer-to-peer network. *Proceedings of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)*, Feb 2004.

[6] B. Cohen. Incentives build robustness in BitTorrent. *Workshop on Economics of Peer-to-Peer Systems*, Jun 2003.

[7] J.R. Douceur. The Sybil attack. *International Workshop on Peer-to-Peer System(IPTPS'02)*, Mar 2002.

[8] E. Frost. Postmortem [RedHat 9 Binary ISO download on BitTorrent]. http://f.scarywater.net/postmortem/.

[9] P. Golle, K. Leyton-Brown, I. Minronov, and M. Lillibridge. Incentives for Sharing in Peer-to-Peer Networks. *Proceedings 3rd ACM Conference on Electronic Commerce (EC-2001)*, Oct 2001.

[10] O. Kim and M. Walker. The free rider problem: Experimental evidence. *Public Choice*, 1984.

[11] H. T. Kung and C. Wu. Differentiated Admission for Peer-to-Peer systems: Incentivizing Peers to Contribute Their Resources. *Workshop on Economics of Peer-to-Peer Systems*, Jun 2003.

[12] G. Marwell and R. Ames. Experiments in the provision of public goods: I. resources, interest, group size, and the free-rider problem. *American Journal of Sociology*, 1979.

[13] R. C. Merkle. Protocols for public key cryptography. *Proceedings of the IEEE Symposium on Security and Privacy*, Apr 1980.

[14] T. Ngan, A. Nandi, and A. Singh. Fair Bandwidth and Storage Sharing in Peer-to-Peer Networks. *First IRIS Student Workshop*, Aug 2003.

[15] T. Ngan, D. Wallach, and P. Druschel. Enforcing fair sharing of peer-to-peer resources. *International Workshop on Peer-to-Peer System(IPTPS'03)*, Feb 2003.

[16] H. E. Peters, A. S. Unur, J. Clark, and W. D. Schulze. Free-Riding and the Provison of Public Goods in the Family: A Laboratory Experiment. *International Economic Review*, 45:283–299, Feb 2004.

[17] K. Ranganathan, M. Ripeanu, A. Sarin, and I. Foster. "To Share or Not to Share" an analysis of Incentives to Contribute in Collaborative File Sharing Environments. *Workshop on Economics of Peer-to-Peer Systems*, Jun 2003.

[18] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. *Proceedings of Multimedia Computing and Networking*, Jan 2002.

[19] C. Shannon and D. Moore. The Spread of the Witty Worm. http://www.caida.org/analysis/security/witty/, Mar 2004.

[20] J. Shneidman and D. C. Parkes. Rationality and Self-Interest in Peer-to-Peer networks. *2nd International Workshop on Peer-to-Peer Systems (IPTPS 2003)*, Feb 2003.

[21] V. Vishnumurthy, S. Chandrakumar, and E.G. Sirer. Karma : A secure economic framework for peer-to-peer resource sharing. *Workshop on Economics of Peer-to-Peer Systems*, Jun 2003.