Leveraging social networks for increased BitTorrent robustness

Wojciech Galuba, Karl Aberer
Ecole Polytechnique Fédérale de Lausanne (EPFL)
firstname.lastname@epfl.ch

Zoran Despotovic, Wolfgang Kellerer
DOCOMO Euro-Labs, Munich, Germany
lastname@docomolab-euro.com

***Abstract*—In peer-to-peer content delivery systems, such as BitTorrent, there may exist nodes that are non-cooperative and do not contribute their upload bandwidth to the system while still downloading content from others. The current widely used countermeasures against this freeriding behavior have been shown to be ineffective. In this paper, we address the problem by leveraging the trust latent in the social networks and explicitly incorporating the social links as part of the BitTorrent content distribution infrastructure.**

Our extensive system evaluation produces several insights. First, the social network topology alone without the trackers is an efficient and scalable content distribution medium. Second, thanks to the cooperative social links, BitTorrent's robustness to freeriding significantly improves. Finally, we find that a hybrid solution in which peers download from both their friends and other peers obtained from the trackers has the highest robustness to freeriding, shortest download completion times and the most balanced upload bandwidth utilization.

Keywords: P2P networks, content distribution, incentive mechanisms

I. Introduction

Most peer-to-peer content delivery systems are designed to be open, any peer can join and any peer can potentially communicate with any other peer in the system. The correct functioning of any P2P system relies on cooperation between the peers. In large-scale open systems peers have little prior knowledge about one another and cannot assume that the others will always cooperate. BitTorrent peers can, for example, choose to download files but not upload any to conserve bandwidth. To prevent the freeriding behavior, additional mechanisms have to be built into the system. The design of such mechanisms is a non-trivial problem and has received considerable attention ([21], [14], [10], [16]). We propose an alternative solution to the problem based on the following observation.

In most of the widely-deployed P2P content delivery systems a single peer usually corresponds to a single user. Given the popularity of some of the P2P applications, it is very likely that for most users a significant fraction of their social network friends also use the same application. Most P2P content delivery systems do not take advantage of these social links.

In a ***social P2P system***, communication occurs almost exclusively between nodes whose owners have a trustworthy social relationship. These systems are also known as darknets [3], [19] or friend-to-friend (F2F) systems [13]. The basic assumption behind such systems is that just as people interacting in the real world are much less likely to cheat against friends than strangers, peers in an F2F system are likely to behave cooperatively towards other peers knowing that they belong to friends. There is thus little need for additional cooperation-enforcing mechanisms. This simplifies the system design.

In this paper we evaluate a socially-aware BitTorrent, in which peers connect both to the peers obtained from the trackers and to the peers belonging to the user's friends. The three core problems that we address are: (1) whether the social network topology alone can function as a scalable and efficient content distribution medium, (2) whether we can take advantage of the social links for solving the freeriding problem without sacrificing performance and (3) what is the number of downloaders in the social network at which the benefit of using the social links becomes significant.

In the first system variant we consider communication occurs strictly only along the social links. The social network topology alone turns out to be an efficient content distribution medium (§IV-B) with the same system scaling properties as the tracker-based BitTorrent (§IV-F). Only a small poorly connected minority of the peers finish late, which is unavoidable due the social networks' power-law node degree distributions [12].

We also examine the BitTorrent's tit-for-tat (TFT) mechanism [5] in the presence of freeriders. As in the other studies ([21], [14], [10], [16]), our measurements confirm that TFT is not an effective solution to the freeriding problem. On the other hand, the social BitTorrent turns out to be remarkably robust to freeriding (§IV-B). There is only a slight increase in completion times even as 33% of the peers are freeriders, while in the classical tracker-based BitTorrent, the impact of the freeriders on performance is much more significant.

One caveat of relying only on the social links for content distribution is that not all the friends in the social network might be interested in downloading the same content. If only the social links are used for the downloads and the content is rare, not all the peers might find download partners among their friends and may not be able to complete the download. For rare content, peers must thus still rely on the trackers to discover the other peers interested in the same content. We evaluate a hybrid design in which peers use both their friends and the peers from the trackers for downloads. We show that in such a system, even if the content is rare the social links still improve the robustness to freeriding (§IV-D). The social BitTorrent's downloads are faster than regular BitTorrent's up until the fraction of downloaders in the social network falls below 2%. The 2% threshold clearly delineates the setups in which the social networks can be leveraged to combat the freeriding in BitTorrent. Our measurements additionally show that apart from solving the sparse swarm problem, the hybrid solution has the most balanced upload bandwidth utilization (§IV-E) among all the evaluated system variants.

II. Related work

There are several P2P systems that use the concept of social networks in one way or another. Li et al. [13] propose an F2F system for scalable and durable storage. The Freenet project [4] turns a social network into a privacy-preserving file-sharing application with focus on efficient search [20]. Tribler [18], another file-sharing application, allows the creation of social groups. Peers can recruit their friends from the group as download helpers to improve the performance. The helpers contribute their bandwidth twice, downloading from the source and then uploading to the friend that requested help. In contrast, in our system there is no recruitment but we do take advantage of the fact that two friends are downloading the same content. None of the above work looks at the social network topology as the sole means of content distribution and does not clearly measure the performance benefits gained from using the social links.

The popular BitTorrent client, Vuze, has recently added the friends feature[1]. Users can allocate more upload bandwidth to their friends and easily share torrent links with them. The eMule client [11] has friend lists as well. Users can allocate upload slots to friends so they do not need to wait in the queue. However, up to our knowledge the performance impact of the friend slots has not been investigated for large-scale BitTorrent swarms, we do that in this paper.

The original solution to the problem of freeriding in BitTorrent is the tit-for-tat algorithm which has been shown to have limited effectiveness ([21], [14], [10], [16]). Several other solutions have been proposed. Locher et al. [15] use coding to increase block diversity in the network and change the seeder strategy to only upload a globally fixed set of blocks per leecher. This forces the freeriders to upload blocks to other peers in order to obtain new blocks in return. In the Piatek's et al. approach [17], peers share download accounting information (bytes sent, received etc.) of their neighbors with the other neighbors. This one-hop reputation system enables indirect reciprocity as opposed to TFT's pairwise reciprocity which results in increased performance. We propose to solve the problem of freeriding by taking advantage of the social links, for which cheating is unlikely. Each peer's identity is tied to a real person's identity and we, in a way, use the external ”human reputation system” for ensuring cooperative peer behavior (§III-C).

Social networks can be used to improve the security and privacy of peer-to-peer systems by constraining the communication only to the social links and cryptographically protecting the exchanged data [19]. Friends can authenticate one another, for example with PGP or other methods [2]. One such system is OneSwarm [9], a friend-to-friend content delivery system based on the Azureus BitTorrent client that leverages the social network of Google Talk users to build an F2F system with the goal of inhibiting the monitoring of user behavior by external parties. The nodes communicate strictly only along the social links. Content search is done by flooding the social network with requests. The returning search responses build paths in the network along which the content is then forwarded hop-by-hop. This results in a much higher bandwidth consumption, proportional to the number of hops. In contrast, in our system all content is shared in a single hop as in the original BitTorrent and we take advantage of the social network to solve the freeriding problem and do not address the privacy or security.

III. System architecture

Making BitTorrent social requires only a few changes. In particular, the content distribution protocol does not have to be modified in any way, we only change the way in which peers discover one another. We also leave the standard mechanisms unchanged, such as the periodic chokes/unchokes, the end-game protocol and optimistic unchoking.

A. Friend connections

Since the social network topology is the integral part of the system we need to provide the users with a way to manage their social links (friend lists). To create a new social link the two users exchange cryptographic information that is later used for mutual authentication. Existing infrastructures such as PGP or F2F-specific protocols [2] can be used to facilitate that. For additional security the friend-to-friend communication can be encrypted.

When a peer starts up, it needs to be able to find the peers belonging to the other friends and connect to them. There exist several ways of achieving this, most of the instant messaging clients have solved this problem (e.g. the open decentralized Jabber/XMPP[2] infrastructure). The social BitTorrent could also be integrated with the existing social network platforms using their APIs, e.g. Facebook[3]. In our evaluation we simply use a single instance of a friend server through which the peers discover one another.

The connections to friends are added to the local peer list and the standard BitTorrent protocol is executed as if these peers were discovered through a tracker.

B. Friend upload priority

In the social BitTorrent the friend connections are treated specially. Every time the upload slots are allocated, which happens periodically every 10s in our implementation, the friend connections take absolute priority over the other connections irrespective of their upload and download histories. For other, non-friend connections we are still using the **tit-for-tat (TFT)**, i.e. upload slots are allocated to the peer that is uploading the most.

[1] `http://azureus.sourceforge.net/`

[2] `http://xmpp.org/about/jabber.shtml`

[3] `http://developers.facebook.com/`

C. Freeriding disincentive

The users are made aware of the friend connections and all the activity occurring on them. The status of the friend connections and the friend identities are displayed on the user interface. If some friend refuses to upload anything or artificially caps the upload bandwidth, it is clearly visible and the identity of the cheater is known. We assume throughout the paper that this provides a strong disincentive for freeriding on friends, but does not prevent the users from freeriding on non-friend peers.

IV. Evaluation

We have implemented the social BitTorrent in the ProtoPeer framework[4] based on the unofficial BitTorrent specification[5] complete with the trackers, the tit-for-tat mechanism, optimistic unchoking, partially downloaded piece prioritization and the end-game strategy.

We use the flow-based network model[6] in ProtoPeer which assumes a fixed upload bandwidth cap for each peer and divides the bandwidth equally among the outgoing flows. For simplicity, we assume the download bandwidth to be unbounded, since the upload bandwidth is typically the bottleneck in file-sharing systems. To introduce diversity, we assume that 10% of the peers have a high 2Mbit/s upload bandwidth, 30% are with 1Mbit/s and the rest with 512kbit/s, which is loosely based on the residential broadband measurements in [6]. In all the experiments we assume that initially a single source has the complete file. The source has a 2Mbit/s upload bandwidth.

The social network topologies for our evaluation were generated based on node degree and clustering coefficient distributions of an actual large-scale social graph of a popular instant messaging client [12]. The node degree is bounded by 64 and the graph is connected.

Throughout our evaluation we assume that the peers arrive in the system within the first 10 seconds and never depart. We have determined in simulations that peer arrivals and departures affect the social BitTorrent in the same way as the regular one and we omit these results.

Unless otherwise stated, the downloaded content is a 25MB file divided into 50 512kB pieces, each consisting of 32 16kB blocks. All peers use the rarest first piece selection strategy. The network consists of 256 nodes including the initial source. We vary the network size and the content size in separate scalability tests (§IV-F).

The ***freeriders*** in the system are defined as peers who ignore all requests for content, but still request the content from the other peers.

A. The lineup

We consider the following systems: **BT** - the classical BitTorrent system using a centralized tracker to discover 25 random peers to connect to, **SBT** - the strictly social variant of BitTorrent running on top of the generated social network, the data flows only along the social links, no trackers are used and **HBT** - the hybrid BitTorrent system, which uses both the social links and the trackers to discover additional uploaders as in the original BitTorrent.

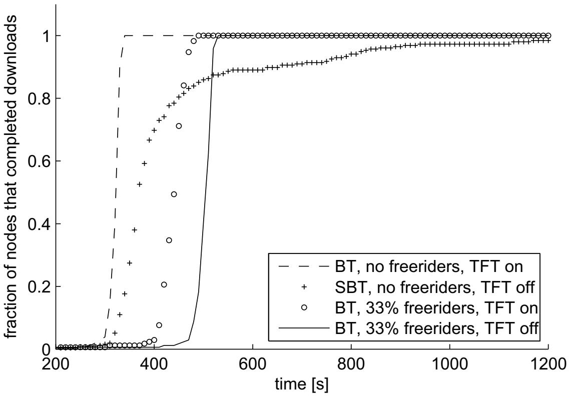


Fig. 1. **Completion times.** A 25MB file is distributed from a single initial seeder using the BitTorrent system (BT) and social BitTorrent (SBT). We measure the download completion times at non-freerider peers. The first 70% of the peers in SBT complete their downloads 1 minute later than in BT. The next 30% of the peers in SBT are mainly the nodes that are poorly connected and require more time for the download. When every third peer is a freerider in BT, the downloads slow down by 200s. Enabling tit-for-tat (TFT) improves the performance, but BT is still slower than SBT.

B. Completion time

The most important performance metric of a file-sharing system is the time it takes to complete the file download on all the peers (Fig. 1).

Content distribution in the social network is not significantly slower than in the open tracker-based system. However, SBT suffers from the long-tail problem: the last 20% – 30% of the peers complete their downloads much later than the majority. Social graphs have a power-law degree distribution [12] with many peers having only a few links. It is those peers that cause the problem, they do not have enough peers to download from and the whole download process is delayed.

The set of results on Fig. 1 also illustrates the superior robustness to freeriding of the social BitTorrent. When freeriders are present, the open system (BT) is slower than the social system (SBT) for the majority of the peers event when tit-for-tat is enabled.

C. Sparse swarms

Up to now we have assumed that all the peers in the social graph are interested in downloading the same file. This is not likely in reality. Peers that are not interested in the file do not download or upload any data. Let ***swarm density*** be the fraction of peers in the social network that are interested in the file. If the density is low then some interested peers may not be able to reach any seeder via a chain of interested peers and will not complete their downloads. We have confirmed that this effect is in fact severe (Fig. 2).

To solve the sparse swarm problem the peers need some method of finding other peers interested in downloading the

[4] `http://protopeer.epfl.ch`

[5] `http://wiki.theory.org/BitTorrentSpecification`

[6] `http://protopeer.epfl.ch/wiki/FlowBasedNetworkModel`

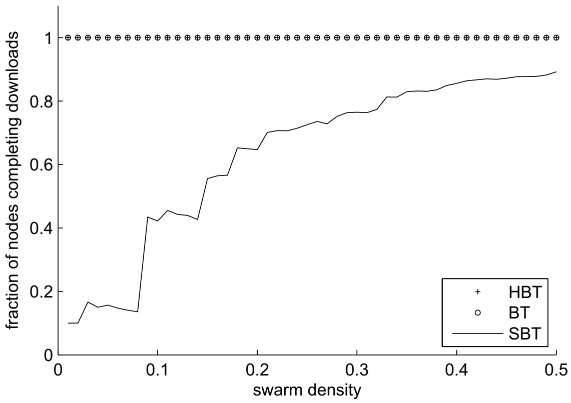


Fig. 2. **Sparse swarms.** In the social BitTorrent system only some of the social network's nodes may want to download the file. This creates a problem: not all the nodes can access a seeder via a chain of uploaders.

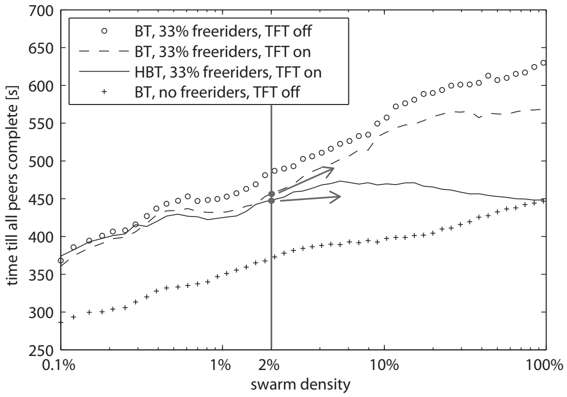


Fig. 3. **Varying the swarm density.** In HBT, peers use both the peers discovered via the tracker and their direct friends as the uploaders. The BitTorrent systems without TFT and without freeriders are included as the baselines. In all systems the social network size is 10000 and the file is 25MB. The performance improvement gained by using the social links (HBT) is the bigger the denser the swarm is. However, when the density falls below 2% (i.e. less than 200 downloaders among the 10000 peers), the performance advantage offered by the social links disappears.

same content. This can either be done in a decentralized way by e.g. flooding the social network with search requests (as in OneSwarm [9]) or by using the trackers as in the original BitTorrent.

D. Adding the trackers - the hybrid system

To complete their downloads in a sparse swarm, peers can discover additional uploaders through a tracker. At the same time, the peers can still use their friends as reliable uploaders when possible. We look at how the performance of such a hybrid system (HBT) changes with the swarm density (Fig. 3).

In high density swarms peers have no trouble finding friends interested in the same file. As the swarm density decreases, peers can no longer rely on downloads from friends and have to download from strangers, which can potentially be freeriders. When the swarm density falls down to approximately 2% the performance advantage of the social BitTorrent disappears, the majority of downloads are from strangers. This clearly shows in which setups the social networks can be used to improve BitTorrent's robustness to freeriding.

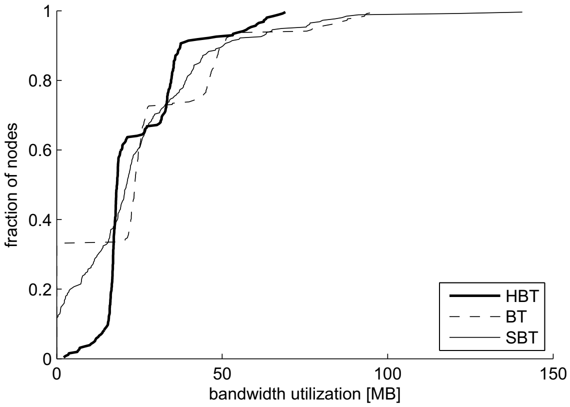


Fig. 4. **CDF of the bandwidth utilization.** A 25MB file is distributed from a single initial seeder in the regular (BT), the social (SBT) and the hybrid (HBT) BitTorrent system. Every third node is a freerider, tit-for-tat is enabled. We measure the cumulative distribution function of the total upload bandwidth used per node during the file distribution. The area to the left of the curve remains constant across the systems, the same amount of data is uploaded in total. The closer the curve is to a vertical line the more balanced the bandwidth utilization is. Note: 10% of the peers have a 2Mbit/s upload bandwidth, 30% 1Mbit/s and the rest 512kbit/s, which is visible on the plots as "steps".

E. Bandwidth utilization

Next we look at how balanced the bandwidth utilization is in the system (Fig. 4). In the case of BT, the 33% of the freeriders are clearly visible, they do not contribute any bandwidth. In the case of SBT, a few socially well connected nodes contribute a disproportionate amount of bandwidth to serving the poorly connected nodes. The hybrid system (HBT) is the most balanced, poorly connected nodes obtain additional download sources from the trackers, while the majority of the freeriding is prevented by leveraging the social links.

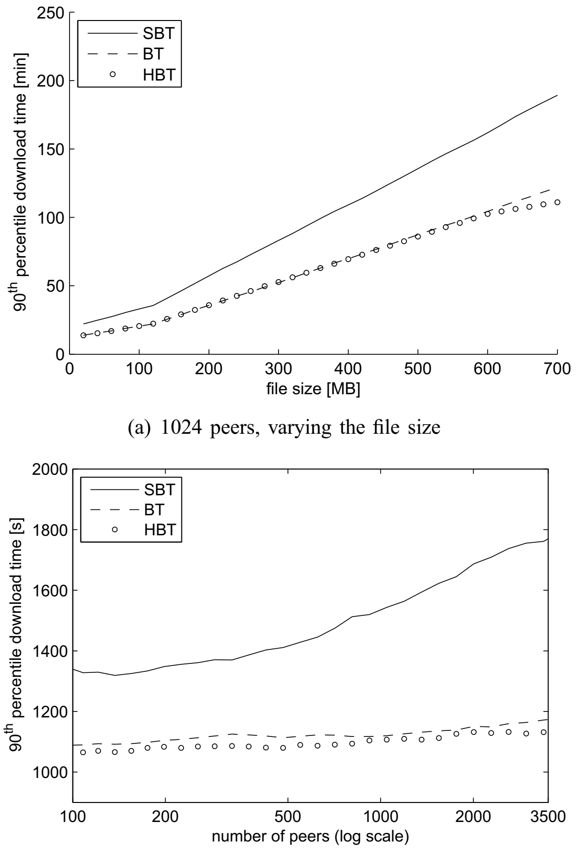
F. Scalability

In all our previous experiments the file size and the network size were fixed. We now examine how the performance changes as these parameters change (Fig. 5).

The results show that the download times scale linearly with the file size and logarithmically with the number of peers in the network. This is in line with the scaling formulas derived in [8] for the optimal BitTorrent content distribution. The social network alone (SBT) turns out to be just as scalable as the classical solution based on the centralized tracker (BT), albeit the completion times are slower due to the power-law degree distributions of the social graphs (§IV-B).

V. Discussion & Conclusions

Social network as a content distribution medium. We have shown that the topology of the social network alone without relying on the trackers can function as an efficient and scalable content distribution medium. For the majority of the nodes in the strictly social BitTorrent using only direct friends as uploaders the completion times are as short as



(a) 1024 peers, varying the file size

(b) 100MB file, varying the network size

Fig. 5. **System scaling.** Every peer in the social network is interested in the file. We vary the network and file size and measure the 90^{th} percentile of download times. All peers are active downloaders. The scaling is linear in the file size and logarithmic in the network size. The system scaling properties of the strictly social BitTorrent are not significantly different than those of the regular tracker-based system.

in a BitTorrent system using trackers to interconnect the peers. However, the peers with fewer friends experience longer completion times and to improve their performance need to find additional uploaders. The best uploaders are simply new friends. This creates a strong incentive for the friends to connect to other friends and could potentially drive the growth of BitTorrent's social network.

Robustness to freeriding. The BitTorrent's tit-for-tat mechanism is known to be an ineffective solution to the freeriding problem ([21], [14], [10], [16]). We have confirmed this in our measurements (§IV-B). When freeriders are present, enabling tit-for-tat increases the performance, but only slightly. This ineffectiveness, as others have observed [1], is mainly attributable to the altruism of the seeders, which unchoke downloaders without regard to their past upload performance.

We have confirmed experimentally that indeed the reliable social links are crucial to increasing the robustness to freeriding. Even when 33% of the additional uploaders are freeriders the performance of the social BitTorrent drops only insignificantly. The social network forms a backbone for reliable cooperative content distribution and the freeriders' impact is lowered.

Best of two worlds: the hybrid system. The BitTorrent's tit-for-tat mechanism does not prevent freeriding, but merely provides a disincentive for it. The freeriders do not contribute any of their upload bandwidth and, as we have measured in §IV-E, the downloaders must download from other peers, which creates a bandwidth utilization imbalance in the system.

In the BitTorrent variant that uses only the social links for downloads (SBT), a small fraction of the peers are poorly connected and use the highly connected peers as download hubs. This is a fundamental problem caused by the power-law node degree distributions of the social graphs [12]. Again, there is a bandwidth utilization imbalance in the system.

As we have shown, the above two approaches can be combined into one (HBT), in which peers connect to both their friends as well as strangers discovered via the centralized trackers. This results in a system with a balanced bandwidth utilization, high robustness to freeriding and short download completion times.

References

[1] L. G. Alix L.H. Chow and V. Misra, "Improving BitTorrent: A Simple Approach," in *Proc. of IPTPS*, 2007.

[2] J. Bethencourt, W. Low, I. Simmons, and M. Williamson, "Establishing darknet connections: an evaluation of usability and security," in *Proceedings of the 3rd symposium on Usable privacy and security*. ACM Press New York, NY, USA, 2007, pp. 145–146.

[3] P. Biddle, P. England, M. Peinado, and B. Willman, "The Darknet and the Future of Content Distribution," in *Proceedings of the 2002 ACM Workshop on Digital Rights Management*, 2002.

[4] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley, "Protecting free expression online with freenet," *IEEE Internet Computing*, vol. 6, no. 1, pp. 40–49, 2002.

[5] B. Cohen, "Incentives Build Robustness in BitTorrent, 2003," 2003.

[6] M. Dischinger, A. Haeberlen, K. Gummadi, and S. Saroiu, "Characterizing residential broadband networks," in *IMC'07*, 2007.

[7] A. Fast, D. Jensen, and B. Levine, "Creating social networks to improve peer-to-peer networking," in *SIGKDD*, 2005.

[8] P. Ganesan and M. Seshadri, "On Cooperative Content Distribution and the Price of Barter," in *ICDCS*, 2005.

[9] T. Isdal, M. Piatek, A. Krishnamurthy, and T. Anderson, "Friend-to-friend data sharing with oneswarm," UW-CSE, Tech. Rep., 2009.

[10] S. Jun and M. Ahamad, "Incentives in BitTorrent induce free riding," in *Proc. of the 2005 ACM SIGCOMM workshop on Economics of P2P systems*. ACM New York, NY, USA, 2005.

[11] Y. Kulbak and D. Bickson, "The eMule Protocol Specification," *eMule project, http://sourceforge. net*.

[12] J. Leskovec and E. Horvitz, "Worldwide Buzz: Planetary-Scale Views on an Instant-Messaging Network," in *WWW*, 2008.

[13] J. Li and F. Dabek, "F2F: Reliable storage in open networks," in *IPTPS'06*, 2006.

[14] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer, "Free Riding in BitTorrent is Cheap," in *Connections*, vol. 300, no. 400, p. 500.

[15] T. Locher, S. Schmid, and R. Wattenhofer, "Rescuing Tit-for-Tat with Source Coding," in *P2P'07*, 2007.

[16] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in BitTorrent," in *NSDI*, 2007.

[17] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson, "One hop reputations for peer to peer file sharing workloads," *NSDI*, 2008.

[18] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. Epema, M. Reinders, M. van Steen, and H. Sips, "Tribler: a social-based peer-to-peer system," *Concurrency and Computation*, vol. 20, no. 2, p. 127, 2008.

[19] M. Rogers and S. Bhatti, "How to disappear completely: A survey of private peer-to-peer networks," in *SPACE 2007*, 2007.

[20] O. Sandberg, "Distributed Routing in Small-World Networks," in *Proc. of the Eighth Workshop on Algorithm Engineering and Experiments*. Society for Industrial Mathematics, 2006.

[21] M. Sirivianos, J. Park, R. Chen, and X. Yang, "Free-riding in BitTorrent Networks with the Large View Exploit," in *Proc. of IPTPS*, 2007.