

# Systemic risk and user-level performance in private P2P communities

Adele L. Jia, Rameez Rahman, Tamás Vinkó, Johan A. Pouwelse, and Dick H.J. Epema.

**Abstract**—Many peer-to-peer communities, including private BitTorrent communities that serve hundreds of thousands of users, utilize credit-based or sharing ratio enforcement schemes to incentivize their members to contribute. In this paper, we analyze the performance of such communities from both the system-level and the user-level perspectives. We show that both credit-based and sharing ratio enforcement policies can lead to system-wide “crunches” or “crashes” where the system seizes completely due to too little or to too much credit, respectively. We present a theoretical model that identifies the conditions that lead to these system pathologies and we design an adaptive credit system that automatically adjusts credit policies to maintain sustainability. Given private communities that are sustainable, it has been demonstrated that they are greatly oversupplied in terms of excessively high seeder-to-leecher ratios. We further analyze the user-level performance by studying the effects of oversupply. We show that although achieving an increase in the average downloading speed, the phenomenon of oversupply has three undesired effects: long seeding times, low upload capacity utilizations, and an unfair playing field for late entrants into swarms. To alleviate these problems, we propose four different strategies, which have been inspired by ideas in social sciences and economics. We evaluate these strategies through simulations and demonstrate their positive effects.

**Index Terms**—Private community, sharing ratio enforcement, credit policy, systemic risk, demand and supply.



## 1 INTRODUCTION

In decentralized collaborative systems, including peer-to-peer (P2P) systems, providing incentives for user to contribute is essential. The well-known P2P file-sharing protocol BitTorrent owes its success to its Tit-For-Tat (TFT) incentive policy, which works reasonably well in fostering cooperation among downloaders (also known as *leechers*). However, TFT does not provide incentives for peers to remain in the system after their downloads are complete in order to *seed* the entire file. Therefore, peers are free to engage in “Hit and Run” behavior, leaving immediately upon completing their downloads.

To provide an incentive for seeding, in recent years there has been a proliferation of so-called *private* BitTorrent communities. These communities employ private *trackers* that maintain centralized accounts and record the download and upload activity of each user. They apply policies to incentivize good overall upload / download behavior. One such well-known policy is *Sharing Ratio Enforcement* (SRE), in which each member is required to keep its *sharing ratio* (the ratio between its total amounts of upload and download) at least equal to a threshold called the *SRE threshold*, which is set by the

community administrator. Community members whose sharing ratios drop below the threshold are first warned and then banned from downloading, or even expelled from the community. Another such policy is the *credit-based* policy, which requires each member to maintain a positive *credit* (its total amount of upload minus its total amount of download). In this paper we explore both the system-level dynamics and the user-level performance in communities adopting such policies.

Considering a private community as an economic system, we analyze its system-level dynamics by studying its potential systemic risk. In economics, systemic risk is the risk of a collapse of an entire economic system or market [13]. We find that in private communities, too much credit distributed too evenly leads to a *crash* in which peers hold abundant credit and are not willing to contribute. Hence, the system seizes to zero throughput containing only leechers. Conversely, too little credit distributed over the peers leads to a *crunch* in which peers do not have enough credit to download, leading to a seized system containing only seeders<sup>1</sup>.

Even when crashes or crunches do not occur, i.e., when the system is *sustainable*, this only ensures that the system is able to function, but not how well it functions. Though many measurement studies [7], [15], [16], [21] have shown that the SRE-based and credit-based policies

- 
- A.L. Jia, J.A. Pouwelse and D.H.J. Epema are with the Parallel and Distributed Systems Group, Delft University of Technology, the Netherlands. E-mail: adele.lu.jia@gmail.com.
  - T. Vinkó is with the Computational Optimization Department, University of Szeged, Hungary.
  - R. Rahman is with the Faculty of Computer Science, Information Technology University, Punjab, Pakistan.

1. A real world example of the crash and crunch is the story of the Capitol Hill Baby Sitting Co-op [11], which was a group of parents who agreed to cooperate to babysit. A crunch happened when most people wanted to save up coupons: they looked for an opportunity to babysit but there was little demand. Later when more coupons were issued a crash happened: most people felt they had enough coupons so they did not want to babysit, leaving the system with huge demand but no supply.

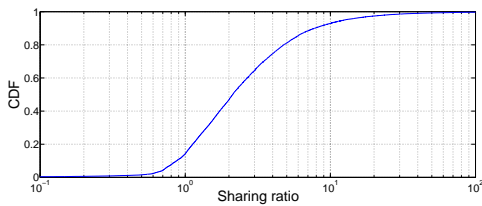


Fig. 1. The CDF of the sharing ratios in CHDBits.org.

are very effective in boosting contribution levels in terms of high seeder-to-leecher ratios and the corresponding high downloading speeds, we argue that the abundant supply of bandwidth also has several negative effects such as excessively long seeding times that are often unproductive. To explore this, we analyze the user-level performance in sustainable private communities.

The main contributions of this paper are:

1. We demonstrate that in private communities credit crashes and crunches can occur and we identify conditions that lead to these extreme outcomes (Section 4);

2. We present a theoretical model that predicts whether a system will crash, crunch, or be sustainable over a defined time horizon. Based on this model we propose an adaptive credit policy that helps the system to avoid crashes and crunches (Section 5);

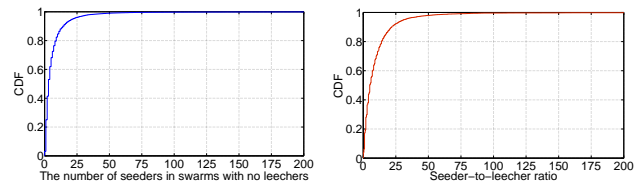
3. We show that users in sustainable private communities achieve high downloading speeds but are forced to seed for excessively long times, during which their upload capacity utilizations are quite low (Section 6). Further, when the popularity of a swarm decreases over time, peers that join the swarm not early enough will have to seed much longer than peers who join (strategically) at the beginning of the swarm (Section 8);

4. We propose and evaluate four new strategies that alleviate these problems while still maintaining a reasonable system-wide downloading speed (Sections 7 and 8).

We use private BitTorrent communities as an example, but our analysis is applicable to any system that adopts contribution enforcement policies, by generalizing the metrics for determining the credit and the sharing ratio from the upload and download amounts to any metrics representing contribution and consumption.

## 2 REAL WORLD OBSERVATIONS

To support our later analysis, we first present real world observations of two private communities, CHDBits.org [2] and Bitsoup.org [3]. CHDBits and Bitsoup both require the users to maintain sharing ratios larger than the threshold of 0.7. The trackers of CHDBits collect information that is periodically reported by the BitTorrent clients of its users, which is displayed in the form of HTML pages available to only its users. We crawled these trackers in May 2011. For each user in CHDBits, we collected the information on its user profile page including the upload and download amount, the seeding time, and the sharing ratio. For each torrent, we collected the information of its published date, and its numbers



(a) The CDF of the number of seeders in swarms with no leechers (b) The CDF of the seeder-to-leecher ratio in swarms with at least one leecher

Fig. 2. Oversupply in CHDBits swarms.

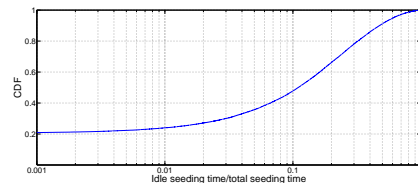


Fig. 3. The CDF of the fraction of idle seeding time of peers with sharing ratios smaller than 1 in BitSoup.org.

of seeders and leechers at the time of snapshot. In total, information on all the 31,547 registered users and 40,040 torrents was obtained. For Bitsoup, we use the traces published in [5] that report the user activity of 84,007 users in 13,741 torrents during a period of two months.

### 2.1 The existence of over-seeding behavior

In CHDBits, maintaining a sharing ratio slightly above the SRE threshold is sufficient for a user to start downloading a new file. However, we observe that not all the users behave like this. As shown in Fig. 1, more than 95% of the users in CHDBits keep sharing ratios higher than 0.7 and more than 50% of the users keep them higher than 2. This phenomenon of peers seeding more than required and achieving sharing ratios that are (much) higher than the SRE threshold has also been observed in many other communities [15].

From the above observation we abstract two user behaviors for our later analysis, *lazy-seeding* and *over-seeding*. *Lazy-seeding* peers seed the minimum amount required by the enforcement policies. They represent the users who are download-oriented, i.e., who only seed enough to maintain adequate sharing ratios or credit to be able to start new downloads. On the other hand, *over-seeding* peers are deposit-oriented, and always maintain sharing ratios (much) higher than required. The behavior of such peers may be triggered by various motivations such as altruism, a desire to be part of the rich elite of the community, or a habit of storing credit for the future. In line with the terminology used in economics, *over-seeding* peers can be understood as *hoarders* as their behavior essentially amounts to hoarding credit.

### 2.2 The oversupply

The main motivation for implementing credit or SRE policies is to close the gap between bandwidth demand

and supply as observed in public BitTorrent communities, where there is significantly more demand than supply [16]. However, the presence of over-seeding peers completely reverses the situation and in private communities, swarms tend to be extremely oversupplied.

At the time of the crawling, CHDBits had 33,041 active swarms (with at least one leecher or one seeder), among which 26,402 swarms (79.9%) had no leechers at all! As shown in Fig. 2(a), 40% of the swarms with no leechers still had at least 5 seeders, and for swarms with at least 1 leecher, the *seeder-to-leecher ratio* (*SLR*) is quite high: as shown in Fig. 2(b), 50% (5%) of these swarms had an *SLR* of at least 6 (30). We see clearly that a majority of the swarms are heavily oversupplied. Therefore, in such swarms, intuitively it is difficult for seeders to perform any actual uploads. We validate our speculation through the following observation.

### 2.3 Unproductive seeding

It is clear that in order to achieve high sharing ratios, peers need to spend considerable amount of seeding time. In the case of over-seeding peers, long seeding times are to be expected. However, we observe that even many peers with small sharing ratios suffer from excessively long seeding times, and a significant part of their seeding time is spent idle without being able to upload anything to others. As a consequence, they have to wait for a long period until their sharing ratios are high enough to start new downloads.

Fig. 3 shows the CDF of the fraction of idle seeding time of peers with sharing ratios smaller than 1 in BitSoup. We see that 10% of these peers spend at least half of their seeding time idle. Note that Fig. 3 only shows the fraction of idle seeding time. It can be conjectured that the fraction of seeding time that is not completely idle yet still yields very low upload speed, would be much higher. We term this situation as *unproductive seeding* and we hypothesize that it is due to the oversupply under credit-based or SRE-based schemes.

## 3 MODEL DESCRIPTION

In this section we explain the credit-based and SRE-based incentive policies, and our model of communities that employ one of these policies.

### 3.1 Credit-based versus SRE-based policies

Credit-based and SRE-based policies are essentially very similar, in a way that they can be understood as variations of each other. The idea behind both policies is that every peer has to maintain a certain relation between the total amount  $u(t)$  it has uploaded and the total amount  $d(t)$  it has downloaded since it entered the community until time  $t$ . The credit-based policy requires users to keep non-negative credit, i.e., to ensure that  $u(t) - d(t) \geq 0$ , while the SRE-based policy requires users to keep a minimum sharing ratio  $SR(t) = u(t)/d(t)$ , i.e.,

to ensure that  $SR(t) \geq \alpha$ , where  $\alpha$  is the SRE threshold. Throughout this paper we assume  $\alpha \leq 1$ , as most private communities do [3], [1]. When  $\alpha = 1$  in the SRE policy, the SRE-based and credit-based policies coincide.

By enforcing non-negative credit in the credit-based policy, the exchanging of data by peers does not generate new credit, and the total amount of credit in the community is always equal to zero (or to the sum of the initial credits allocated to the peers by the community administrator). In contrast, an SRE-based policy allows users to have negative credit (i.e., to have  $u(t) - d(t) < 0$ , which means that  $SR(t) < 1$ ). Holding negative credit increases the amount of credit among the peers with positive credit in the system—in other words, by holding negative credit a user is essentially minting credit. More precisely, the total credit minted by a user in an SRE-based community with  $SR(t) < 1$  until time  $t$  is:

$$d(t) - u(t) = (1 - SR(t))d(t), \quad (1)$$

which is bounded by  $(1 - \alpha)d(t)$ . As the sharing ratios of peers fluctuate, SRE-based communities hold a dynamic amount of credit circulating in the system.

### 3.2 The basic model

We consider a community that is either credit-based or SRE-based. The community comprises a set of  $s$  swarms<sup>2</sup> each associated with a file of size  $F$  (expressed in number of pieces or units), and a set of  $N$  peers each with upload capacity  $U$ .<sup>3</sup> We assume no limit on the download capacity of peers. The download model follows the TFT mechanism in BitTorrent, with seeders uploading units to leechers and leechers exchanging units with each other. In reality, a peer can participate in multiple swarms simultaneously, with its bandwidth shared among all the swarms. However, since the sharing ratio is aggregated over all the swarms, we assume that at any time a peer only participates in one swarm, either as a leecher or a seeder.

The operation of the model is based on *cycles* representing units of time. In every cycle, a peer either uploads and/or downloads data or is idle, and at the end of every cycle, it may switch swarms. Peers attempt to download all  $s$  files in random order.

In a credit-based community, every peer  $p$  is initialized with an amount  $C_p$  of credit, and in an SRE-based community, every peer is initialized with a download amount equal to  $F$  and a sharing ratio that is a uniformly random number between 0 and 2. A peer can and will only start leeching its next file if its credit or its sharing

2. We assume the number of swarms to be large enough that even with no injection of new swarms, users still have enough swarms to download from.

3. Since crashes and crunches are due to credit abundance and shortage, respectively, bandwidth heterogeneity does not influence whether a system will crash or crunch. However, it does influence the user-level performance, for which we examine both bandwidth homogeneous and heterogeneous systems in Section 6.

ratio is at least equal to its *target threshold*, otherwise it continues seeding the current file.

Based on real-world observations, we implement two user behaviors: lazy-seeding and over-seeding (see details in Section 2.1). The target threshold of a lazy-seeding peer is an amount  $F$  of credit in a credit-based community (enough to start and complete leeching a new file) and a sharing ratio equal to the SRE threshold in an SRE-based community.

The condition for a lazy-seeding peer  $p$  at time  $t$  to stop seeding is  $c_p(t) \geq 0$ , with:

$$c_p(t) = \begin{cases} u_p(t) - d_p(t) + C_p - F & \text{credit-based,} \\ u_p(t) - \alpha d_p(t) & \text{SRE-based,} \end{cases} \quad (2)$$

where  $u_p(t)$  and  $d_p(t)$  represent the total amounts of upload and download of peer  $p$ . Over-seeding peers behave in a similar way, but in both credit-based and SRE-based communities they aim at large sharing ratios. Throughout this paper we choose a sharing ratio of 2 as the default target threshold for over-seeding peers. We have run several tests using different values for the threshold and the results show that the tendency of the problem is the same.

## 4 SYSTEMIC RISK: CRASH AND CRUNCH

In this section, we explore the credit dynamics in private communities and we analyze the conditions under which a community will crash, crunch, or be sustainable. We define a *crash* as a situation in which due to credit abundance, peers are not incentivized to contribute and the system completely seizes up, providing no upload or download to any peers. We define a *crunch* as a situation in which due to credit shortages, peers cannot afford new downloads and the system seizes up. We define a community to be *sustainable* if it does not crash or crunch.

### 4.1 Experimental setup

We consider a closed system without new peer arrivals. Peer arrivals bring credit into the system and make it difficult to identify whether the underlying credit dynamics is due to the enforcement policy or to the new credit. In fact, in reality many private communities are (nearly) closed [2], [1]. For example, CHDBits hardly has any open registration and new members can only be admitted by extremely restricted invitation.

The simulation is based on the basic model introduced in Section 3, with  $N = 1000$ ,  $s = 100$ ,  $F = 10$  units, and  $U = 4$  units per cycle. The small file size means the simulation runs produce results at a large scale of granularity. We also performed runs with  $F = 100$  and found no significant difference in results. We choose  $\alpha = 0.7$  as the default value of the SRE threshold, as it is used in many private communities, e.g., [3], [1]. Tests using different values for  $\alpha$  show that the tendency of the problem stays the same, but with different speeds of entering crash or crunch. For each experiment we perform 10 independent runs, and each run is executed for 2000 cycles.

TABLE 1  
Sustainability of the credit-based system.

frac.of rich at start	avg.throughput (std.dev)	avg.frac.of seeders (std.dev)	final state
0.1	0.000 (0.000)	1.000 (0.000)	crunch
0.3	0.218 (0.001)	0.953 (0.005)	sustain
0.5	0.777 (0.002)	0.769 (0.018)	sustain
0.7	0.968 (0.004)	0.506 (0.018)	sustain
0.8	0.587 (0.478)	0.249 (0.204)	sustain/crash
0.9	0.001 (0.000)	0.000 (0.000)	crash

### 4.2 Credit-based: constant credit

As discussed in Section 3, the amount of credit in a credit-based community is always equal to the initial credit allocated by the community administrators. In this experiment, we vary the fraction of peers who are given an initial credit of  $F$  (and other peers are given zero credit), which we call rich peers, thus generating different levels of credit in the system.

Table 1 shows the results of the system containing only lazy-seeding peers. Here, the throughput is expressed as the total amounts of units of data exchanged in the system over an entire run, normalized to the highest one observed in all the experiments. When the fraction of rich peers is initialized to 0.3, 0.5, and 0.7, we see sustainable outcomes with increasing throughput and a smaller number of seeders. We have run extended runs up to 20,000 cycles and find that the sustainable outcomes are maintained. This is intuitive since as the amount of credit in the system increases, fewer peers are poor, and hence more exchange of data can occur.

When 10% of peers are initialized as rich, the system crunches with all peers being seeders; and when 90% of peers are initialized as rich, the system crashes with all peers being leechers. Inspection of individual runs shows that crunches and crashes happen quickly—within the first ten cycles or so. This is reflected in the low (almost zero) throughput under crash and crunch states. It is interesting to see that when the initial fraction of rich peers is set to 0.8, both sustain and crash outcomes can occur. This is reflected in the high variance of the throughput. Here we are very close to the threshold leading to a crash and we find path dependency based on initial random conditions leading to either a high sustainable throughput, or a sudden crash otherwise.

When there is *any* number of over-seeding peers, we find that the system eventually crunches, and the speed of the crunch depends on the number of over-seeding peers. This is intuitive since in our experiments, over-seeding peers seed (to hoard credit) until they have a sharing ratio larger than 2—they eventually hold all the credit in the system and a crunch is inevitable.

### 4.3 SRE-based: dynamic credit

As discussed above, a credit-based community keeps a delicate constant amount of credit which, if not properly set, will lead the system to crunch or crash. On the other hand, as stated in Section 3, an SRE-based system keeps dynamic credit by allowing peers with sharing ratios less

than one to mint some credit. Hence, essentially lazy-seeding peers in an SRE-based community inject credit into circulation, and as in a credit-based community, over-seeding peers absorb credit from circulation.

Intuitively, an SRE-based system cannot be sustainable if all peers are lazy-seeding: soon they will inject too much credit into circulation, which eventually leads the system to a crash. However, with the existence of over-seeding peers who absorb credit from circulation, the effect of credit-injecting by lazy-seeding peers can be alleviated and the system might eventually be sustainable.

We run several simulations to validate the above hypotheses. We consider an SRE-based system in which we vary the fraction of over-seeding peers. As shown in Table 2, a certain fraction of over-seeding peers (0.3 and 0.4 in our experimental settings) does lead the SRE-based community to be sustainable. A too small or a too large fraction of over-seeding peers (0.1 and 0.5 in our experimental settings), on the other hand, eventually leads the system to crash or crunch.

## 5 AVOIDING CRASHES AND CRUNCHES

In this section, we derive (approximate) conditions for predicting crunches and crashes and we design a novel *adaptive credit intervention mechanism* for maintaining the sustainability.

### 5.1 Predicting crashes and crunches

In the model introduced in Section 3, suppose that at time  $t$ , a swarm  $\ell$  has  $x^\ell(t)$  leechers and  $y^\ell(t)$  seeders. Denoting the fraction of the file that a leecher  $x_i^\ell$  still has to download by  $p_i^\ell(t)$ ,  $x_i^\ell$  needs to spend an amount  $\alpha(1 - p_i^\ell(t))F$  of credit to finish its download. We define  $L_i^\ell(t)$  and  $R_i^\ell(t)$  as the sets of peers that have fewer or more pieces of the file than peer  $i$ , respectively (for a seeder  $y^\ell(t)$ ,  $L_i^\ell(t)$  consists of all leechers and  $R_i^\ell(t)$  is empty). We assume that peer  $i$  only downloads from peers in  $R_i^\ell(t)$ , and only uploads to peers in  $L_i^\ell(t)$  (this is not quite true in BitTorrent, which makes the conditions that we will derive approximations). We further assume that the credit paid by peer  $i$  is equally shared by all peers in  $R_i^\ell(t)$ . Hence, if the situation (in terms of  $L_i(t)$  and  $R_i(t)$ ) does not change from time  $t$  onward, peer  $i$  can earn an amount  $Q_i^\ell(t)$  of credit from the peers in  $L_i^\ell(t)$ , where

$$Q_i^\ell(t) := \sum_{j \in L_i^\ell(t)} \frac{(1 - p_j^\ell(t))F}{|R_j^\ell(t)|}. \quad (3)$$

Let  $X_\ell(t)$  and  $Y_\ell(t)$  respectively represent the sets of leechers and seeders that, assuming that the situation does not change from time  $t$ , are able to achieve their target thresholds and start new downloads. Together with Eq. (2), we have:

$$\begin{aligned} X_\ell(t) &:= \{x_i^\ell : c_{x_i^\ell}(t) + Q_i^\ell(t) - \alpha(1 - p_i^\ell(t))F \geq 0\}, \\ Y_\ell(t) &:= \{y_j^\ell : c_{y_j^\ell}(t) + Q_j^\ell(t) \geq 0\}. \end{aligned} \quad (4)$$

TABLE 2  
Sustainability of the SRE-based system.

frac.of over-seeding peers	avg.throughput (std.dev)	avg.frac.of seeders (std.dev)	final state
0.1	0.0093 (0.0012)	0.0000 (0.0000)	crash
0.2	0.2046 (0.2103)	0.0037 (0.0082)	crash/sustain
0.3	0.8910 (0.0041)	0.1487 (0.0141)	sustain
0.4	0.9865 (0.0090)	0.4212 (0.0243)	sustain
0.5	0.1436 (0.0083)	1.0000 (0.0000)	crunch

Now we estimate the remaining download time of leechers and the remaining seeding time of seeders for the current file. Here we assume that during the upload process, leechers and seeders alike upload with their full capacity  $U$  and distribute their upload capacity equally across all the leechers they are uploading to. The estimated remaining download time  $T_i^\ell(t)$  of leecher  $x_i^\ell$  can be expressed as:

$$T_{x_i^\ell}(t) := \frac{(1 - p_i^\ell(t))F}{\sum_{k \in R_i^\ell(t)} \frac{U}{|L_k^\ell(t)|}}. \quad (5)$$

Similarly, the estimated remaining time for a seeder  $y_j^\ell$  to achieve its target threshold and stop seeding is:

$$T_{y_j^\ell}(t) := \max\{0, -c_{y_j^\ell}(t)\}/U, \quad (6)$$

where  $-c_{y_j^\ell}(t)$  (if positive) represents the credit  $y_j^\ell$  still needs to earn to achieve its target threshold.

We can now formulate the *condition for a crunch* to happen in the system as the condition that the sets  $X_\ell(t)$  and  $Y_\ell(t)$  are both empty for all swarms  $\ell$  at some time  $t$ , because that no leechers or seeders are able to earn enough credit to leave their swarms. As a consequence, by the time that the last leecher finishes its download, there will be no exchange of credit in the whole system.

In order to formulate the condition for a crash to happen, let  $P_\ell(t) := \{x_i^\ell : x_i^\ell \notin X_\ell(t)\}$  be the set of leechers in swarm  $\ell$  who will need to seed after finishing their current downloads in order to achieve their target thresholds. Then the *condition for a crash* to happen is:

$$|Y_\ell(t)| = y^\ell(t) \text{ and } \min_{k \in P_\ell(t)} T_{x_k^\ell}(t) > \max_{j \in Y_\ell(t)} T_{y_j^\ell}(t), \quad (7)$$

for all swarms  $\ell$  at some time  $t$ . Note that the system crashes if there are no seeders. The first part of the condition above says that all the seeders in the system will be able to earn enough credit to leave their swarms. If in addition none of the leechers in  $P_\ell(t)$  can finish its download before the last existing seeder leaves the swarm and if this happens to all the swarms (the second part of the condition), then the whole system will end up with no seeders and a credit crash will occur.

### 5.2 Adaptive credit for sustainability

We design a novel *adaptive credit intervention mechanism* to avoid crashes and crunches. At each cycle, we check the conditions for crunches and crashes derived in Section 5.1. When the system is destined for a crunch, it

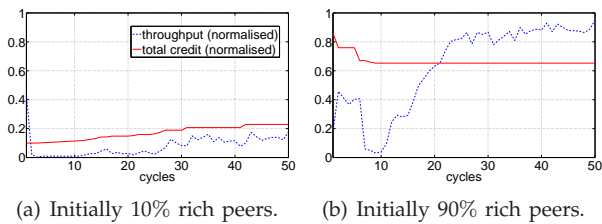


Fig. 4. The normalized throughput and credit in the system with the adaptive credit intervention mechanism.

applies *freeleech*<sup>4</sup> where leechers do not pay any credit for downloading, but seeders and other uploaders are still credited for uploading. Hence, new credit is injected into the system. Credit injection for stimulating the economy has often been used successfully in real world situations [19]. When the system is destined for a crash, it applies *freeseed* where seeding peers (and uploading leechers) do not receive any credit for uploading, but leechers still pay credit for downloading. Hence, credit is removed from the system. We use the credit-based system as an example to evaluate our strategies, but the same analysis can be applied to an SRE-based system. The experimental setup is the same as in Section 4.

In a credit-based system containing only lazy-seeding peers, all runs produce a sustainable outcome, including those previously led to crunches and crashes, i.e., with 0.1 and 0.9 fraction of rich peers at start. Fig. 4.(a) (Fig. 4.(b)) shows how a crunch (crash) is avoided via the activation of *freeleech* (*freeseed*) at several cycles—note the increase (decrease) in credit over time.

As stated in Section 4, any number of over-seeding peers in a credit-based community will eventually lead to a crunch. Fig. 5 shows the throughput in the system with 1% peers being over-seeding—crunches are prevented when the adaptive credit intervention mechanism is applied. However, the throughput of the system in the latter case is still very low, since the adaptive credit intervention mechanism does not attempt to optimize the system. In later sections we provide a more thorough analysis on optimizing the system, i.e., improving the user-level performance.

### 5.3 Discussion

While *freeleech* and *freeseed* policies effectively avoids crunches and crashes, they temporarily suspend the original incentive for contributions, which could lead to reduced performance if users learn to game the system by only downloading during *freeleech* periods and not seeding during *freeseed* periods. A refinement that will help preserve incentives is to reduce the *freeseed* and *freeleech* “tax” amount: rather than having leechers not pay anything at all for downloading and seeders not being credited for uploading, they can be charged or credited for a fraction, say 50%. Further, the taxation

4. *Freeleech* is sometimes also used in existing private communities such as CHDBits, but in a more empirical manner.

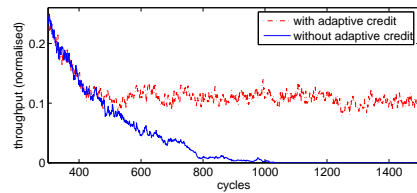


Fig. 5. The normalized throughput with and without the adaptive credit intervention mechanism (50% rich peers).

amount can also be variable and be applied in a continuous fashion. We explore this later in Section 7.

Until now, we have analyzed the sustainability of a P2P community that adopts a credit-based or SRE-based policy. However, the sustainability of the system only ensures that the system is able to function, but does not guarantee it will function well (recall Fig. 5 for an example of a sustainable system with low performance). To explore this, in the following sections we analyze and improve the user-level performance in sustainable P2P communities. There, we take sharing ratio enforcement as an example, but our analysis is also applicable to the credit-based policy, since it is only a special case of SRE with a threshold equal to one.

## 6 USER-LEVEL PERFORMANCE: THE POSITIVE AND NEGATIVE EFFECTS OF SRE

In this section we show the user-level performance under SRE. Based on simulations we examine the influence of several parameters and we exhibit the main reasons for the positive and negative effects of SRE.

### 6.1 Experimental setup

In Section 4 we have shown that in closed private communities crashes or crunches easily happen. It is not worthwhile to analyze the user-level performance in an unsustainable system. Hence, in this section we consider an open system with peer arrivals. As stated in Section 3.1 and 4.3, new peers bring credit into the system and the increase of the credit level alleviates the potential systemic risk. In reality, there are many private communities with open registration, e.g., BitSoup [3], and they can be considered as open systems.

We use the same simulator and consider the same initial settings as in Section 3, except that now we consider 100 initial peers and 5 swarms in the system. In each cycle, new peers arrive according to a certain arrival rate (1 or 10 peers per cycle in our simulations) and they join a random swarm to download. After the first download, they maintain a sharing ratio above their target thresholds. Each peer (with upload capacity 1 unit per cycle) attempts to download all the 5 files (with size of 10 units) in the system, in random order. We consider a bandwidth-homogeneous BitTorrent system unless otherwise indicated. We run the simulation for

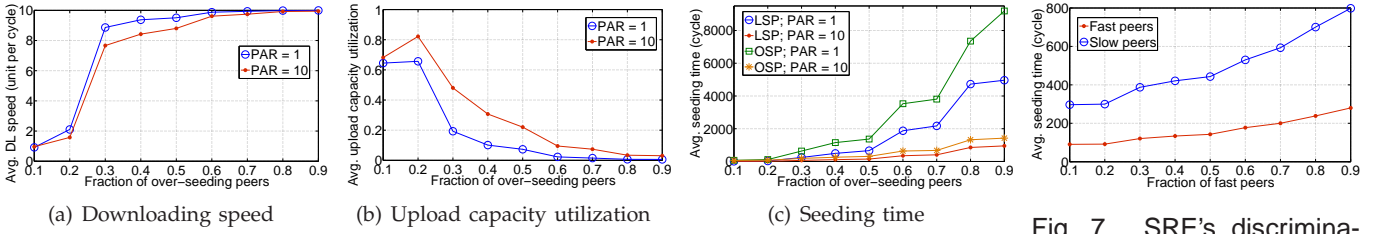


Fig. 7. SRE's discrimination.

Fig. 6. User-level performance under different fractions of lazy-seeding peers (LSP) and over-seeding peers (OSP), and different peer arrival rate (PAR).

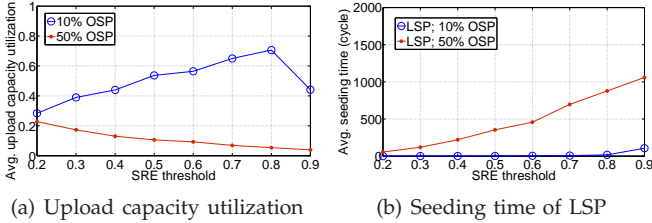


Fig. 8. Influence of the SRE threshold.

2000 cycles and keep a record of peers who finish downloading all the files by the end of the simulation. The results represent the average of 5 runs.

## 6.2 The imbalance of bandwidth supply and demand

In our first experiment we vary the fraction of over-seeding peers, thus generating different levels of over-supply. As shown in Fig. 6(a), with the fraction of over-seeding peers increasing from 0.1 to 0.9, the average downloading speed is increased nearly 10 times. However, the average upload capacity utilization is significantly deteriorated and the seeding time is increased dramatically. With 50% over-seeding peers, on average each peer can only utilize less than 20% of its upload capacity (Fig. 6(b)) and the seeding time of a lazy-seeding peer is nearly 200 times more than its downloading time, and for over-seeding peers, it even increases to over 400 times. On the other hand, with a smaller peer arrival rate (which means a smaller demand) the imbalance and hence the performance, are even worse. As shown in Fig. 6, when the peer arrival rate decreases from 10 to 1 peer per cycle, with the same fraction of over-seeding peers, the average upload capacity utilization is decreased 2-3 times and the average seeding time is increased 2-5 times.

## 6.3 The influence of the SRE threshold

Many communities [3], [1] use 0.7 as the default value of the SRE threshold, empirically or intuitively. This section complements the necessary analysis behind the choice. Fig. 8 shows that with 50% over-seeding peers, when the SRE threshold is increased from 0.2 to 0.9, the upload capacity utilization decreases while the average seeding time increases. Surprisingly, with 10% over-seeding peers, the upload capacity utilization is first

increased and then decreased. We believe this is due to, what we term as, the *seeder's dilemma*: with either a very small or a very large number of seeders, peers cannot well-utilize their upload capacities. The former case is due to the *piece availability problem*: When there are not enough seeders, leechers have to exchange data with each other, which is not always possible since they only hold a part of the entire file. The latter case is due to the insufficient download demand. Without enough demand, though seeders have the will, they cannot find enough leechers to upload to.

## 6.4 The discrimination against peers with limited capacities

In this subsection, we analyze SRE's effects in bandwidth-heterogeneous systems. We simulate a system with slow and fast peers. All the other settings are the same as in previous experiments, except that the upload capacity of fast (slow) peers is 4 (1) units per cycle. We change the fraction of fast peers from 0.1 to 0.9 and the results are shown in Fig. 7. When 30% of the peers are over-seeding, slow peers need to seed 200 to 500 cycles more than fast peers. In general, slow peers need to seed 4 times as long as fast peers, which is the same as the ratio between the upload capacity of a fast and a slow peer. This result is also consistent with our previous theoretical results [10].

While fast and slow peers both put all their effort in participating in the community, slow peers need to seed much longer. We term this as SRE's *discrimination* against low-capacity peers. Together with the long seeding time and the low upload capacity utilization, the user-level performance is deteriorated. In the following section, we propose four strategies to alleviate these problems.

## 7 PROPOSED STRATEGIES

Inspired by ideas in social sciences and economics, in this section we propose four strategies aimed at alleviating the negative effects of incentive policies used in private BitTorrent communities, which require only a minor revision of those policies.

### 7.1 Strategy description

**Negative taxation** is used to describe the supplemental pay from the government to people earning below a certain amount [9]. Under this strategy, the upload amount

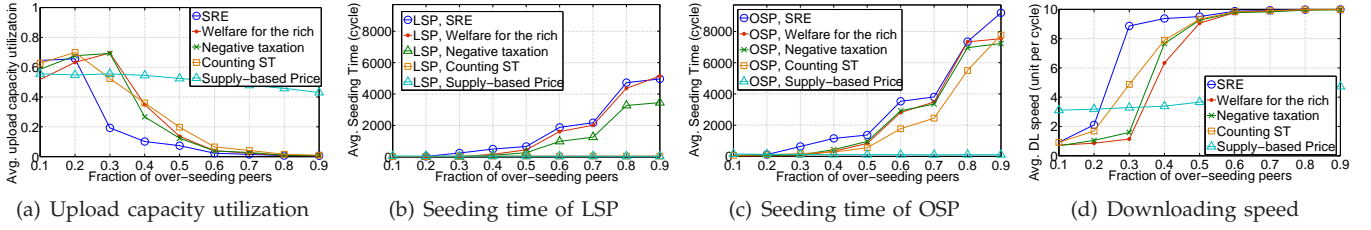


Fig. 9. Strategy performance under different fractions of lazy-seeding peers (LSP) and over-seeding peers (OSP).

of a peer is calculated as its actual upload amount multiplied by the coefficient  $T = \max\{\min\{1/SR, \theta\}, 1\}$ , where  $SR$  represents its sharing ratio and  $\theta$  represents the *maximum negative taxation degree*. In fact, this strategy reduces the SRE threshold individually for each peer to only  $1/T$  of the original one, where  $1 \leq T \leq \theta$ . To still avoid free-riding to a reasonable extent, a small value of  $\theta$  is desirable. In our simulations we set  $\theta = 2$ .

**Welfare for the rich** is used to describe the bestowal of grants and tax-breaks to the wealthy [14]. Under this strategy, the upload amount of a peer is calculated as its actual upload amount multiplied by the coefficient  $\mathcal{W} = \max\{\min\{SR, \varphi\}, 1\}$ , where  $\varphi$  represents the *maximum welfare degree*. In fact, this strategy reduces the desired sharing ratio of an over-seeding peer to  $1/\mathcal{W}$  of its original one, where  $1 \leq \mathcal{W} \leq \varphi$ . In this way, over-seeding peers may achieve their desired sharing ratio and leave oversupplied swarms sooner. To prevent they leave too quickly, a small value of  $\varphi$  is desirable. In our simulation we set  $\varphi = 2$ .

**Remuneration according to effort** is introduced in participatory economics [4], where people are paid according to the effort rather than the amount of contribution. Measuring the effort in terms of the seeding time, we propose SRE with *counting seeding time*, where a peer can start a new download when either it has achieved the SRE threshold or it has seeded for a sufficiently long time, which equals the size of the shared file divided by its upload capacity in our simulation. In this way, peers that are stuck in long seeding process in oversupplied swarms can leave and perform further downloads, which helps to balance the bandwidth demand and supply in the system. Note that since over-seeding peers are deposit-oriented, they still start new downloads only when they have achieved their desired sharing ratios.

**Supply-based price.** Taking inspiration from the law of supply and demand [6], we propose SRE with supply-based price, in which the download amount of a peer is calculated as its actual download amount multiplied by the price  $\mathcal{P} = \max\{1/SLR, \phi\}$ . The price is inversely correlated with the supply in the swarm, which is measure by the *seeder-to-leecher ratio* (SLR). In an oversupplied swarm, a leecher pays less and it is less likely for it to have an insufficient sharing ratio and thus stay as a seeder. In this way, the oversupply is mitigated and the strategy gives a way to escape out of the seeder's dilemma as described in Section 6.3. A similar argument

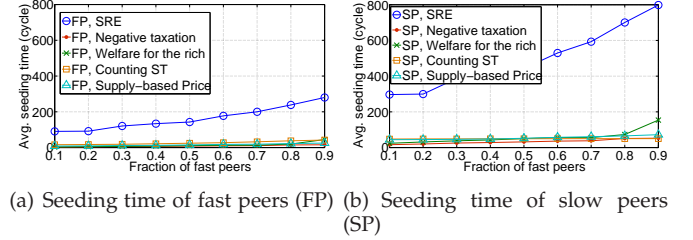


Fig. 10. Alleviating discrimination.

is also applicable in an undersupplied swarm. Here,  $\phi$  represents the *lowest price*, which is used to prevent strategic users to only join swarms with extreme low prices. In our simulation we set  $\phi = 0.1$ . Note that the freeleech strategy proposed in Section 5.2 is an extreme with the price equal to 0.

## 7.2 Strategy evaluation

We evaluate the performance of the new strategies through simulations. The experimental setup is the same as in Section 6 and results are shown in Figs. 9 and 10.

From Fig. 9 we see that by using any of the new strategies, peers achieve higher upload capacity utilizations and shorter seeding times. Particularly, SRE with supply-based price performs stably when the fraction of over-seeding peers increases and it achieves the best performance in reducing the seeding time for all peers, which is reduced by three orders of magnitude. We believe the main reason is that, as stated in Section 7.1, SRE with supply-based price adaptively adjusts the supply and demand in a swarm.

While the seeding time is dramatically reduced, as a trade-off, the average downloading speed is decreased (Fig. 9(d)), hence the downloading time is increased. However, given that in our simulations we consider files with size equal to 10 units, the increase of the downloading time (tens of cycles) is negligible compared to the decrease of the seeding time (hundreds or even thousands of cycles). Further, when the system is consisted by both fast and slow peers, we see from Fig. 10 that *all* the proposed strategies effectively alleviate SRE's discrimination against low-capacity peers: with 30% over-seeding peers, the difference between the seeding times of fast and slow peers are reduced from hundreds to tens of cycles.



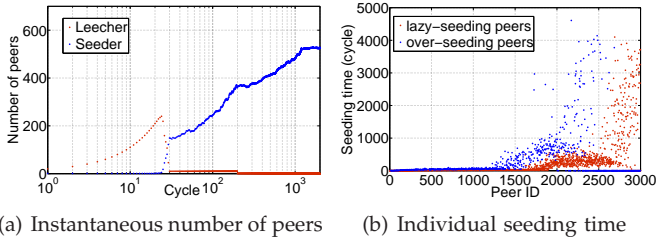


Fig. 11. SRE under dynamic popularity.

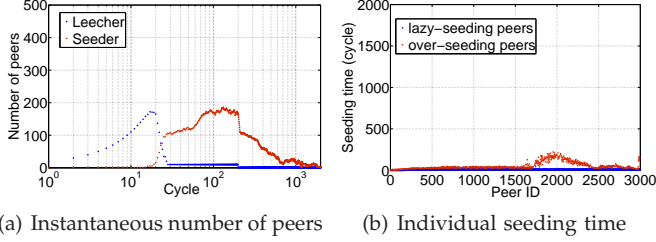


Fig. 12. Counting seeding time under dynamic popularity.

## 8 DYNAMIC FILE POPULARITY

So far, we have only considered scenarios in which all files have the same constant popularity. However, many measurement studies [5], [12] show that the popularity of a file decreases quickly after it is first published. In this section, we analyze the effects of SRE and evaluate our proposed strategies under dynamic file popularity.

### 8.1 Experimental setup

We use the same simulator and consider the same initial settings as in Section 3, except that to better abstract the effects of dynamic file popularity, we only consider one swarm with decreasing popularity. The simulation starts with one injector, who stays in the swarm as a permanent seeder. In successive cycles, new peers arrive according to an exponentially decreasing arrival rate ( $\lambda(t) = \lambda_0 e^{-\frac{t}{\tau}}$ ), a peer arrival pattern that has been observed in many BitTorrent swarms [17]. Each peer joins the swarm with zero upload and download amounts. After a peer finishes its download, it seeds, if necessary, until it achieves its target threshold.

By default, we set 30% peers to be over-seeding. As shown in Section 6, this percentage is typical for showing the effects of SRE. We choose  $\lambda_0 = 10$  and  $\tau = 300$  and we run the simulation for 2000 cycles. All together 3000 peers are included. We have also tested different values for  $\lambda_0$  and  $\tau$ , which give very similar results.

### 8.2 The effects of SRE under dynamic file popularity: arrive sooner to avoid long seeding time

We first demonstrate the effects of SRE under dynamic file popularity. Note that a smaller peer ID means an earlier arrival time. As shown in Fig. 11(a), after the first 30 cycles, the swarm is occupied with hundreds of seeders but only with very few leechers. The presence of

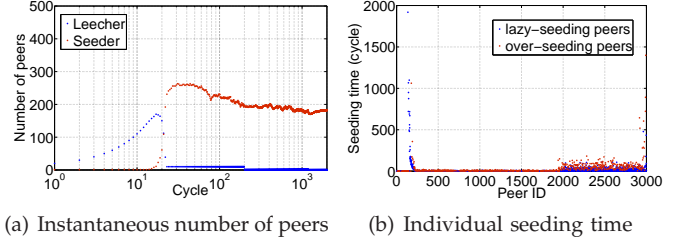


Fig. 13. Supply-based price under dynamic popularity.

existing seeders increases the difficulty for a new seeder to achieve its target threshold and leave the swarm, and vice versa. We term this as *cumulative seeder effect*. As a consequence, peers that arrive later experience much longer seeding times. As shown in Fig. 11(b), to achieve the target thresholds, the first 500 peers only need to seed tens of cycles. After this, the seeding time increases quickly to hundreds or even thousands of cycles.

### 8.3 Proposed strategies under dynamic popularity

As stated in Section 7.2, SRE with counting seeding time and SRE with supply-based price achieve the best performance in alleviating the side-effects of SRE. In this section, we evaluate their performance under dynamic file popularity.

We see from Fig. 12 that SRE with counting seeding time dramatically alleviates the oversupply: except for the large number of seeders during the first 200 cycles, the instantaneous number of seeders is almost always under 100. Without the oversupply, the seeding times of all peers are also dramatically decreased to less than 100 cycles. And as shown in Fig. 13, SRE with supply-based price effectively stabilizes the supply: the instantaneous number of seeders stays stable after 200 cycles. With this constant supply (and not oversupply), peers experience much shorter seeding times compared to adopting the original SRE. Meanwhile, peers who arrive too early or too late have relatively longer seeding times than other peers. The former case is due to the high price for downloading when the swarm is occupied with much more leechers than seeders; in the latter case, few new peers arrive and the lack of demand increase the seeding time.

## 9 RELATED WORK

This paper is based on two previous papers [10], [18] with extensions including demonstrating detailed measurement results, unifying the analysis of SRE-based and credit-based private communities, analyzing the systemic risk of SRE-based private communities, as well as the influence of swarm popularity.

Several P2P incentive schemes based on credits have been proposed in the literature. Vishnumurthy *et al.* [20] present a system involving a virtual currency called *Karma*, whose amount in the system is maintained to avoid inflation and deflation. Kash *et al.* [11] show that

in a scrip system where agents can consume and produce services, both an overabundance of money supply and its shortage lead to inefficiency. They also consider hoarders and how to optimize the credit supply. Our work is different in that we apply multiple user behaviors rather than focusing only on one that optimizes the utility. In addition we focus on detecting and avoiding extreme crashes and crunches.

As stated, for grounding our work we chose the realm of private communities. Chen *et al.* [7] compare 13 private communities with 2 public ones and show their differences in torrent evolution, user behaviors, and content distribution. Liu *et al.* [15] also perform measurement studies and further develop a model to show that SRE indeed provides effective incentives, but is vulnerable to collusion. Andrade *et al.* [5] focus on the dynamics of resource demand and supply, and they show that users typically try to increase their contribution levels by seeding for longer and not by providing more bandwidth. However, our paper shows that providing limited bandwidth is not the will of users, but it is a consequence of the oversupply. Chen *et al.* [8] also notice the oversupply problem and provide a model to identify the optimal stable SLR range. However, they did not propose strategies to solve the problem. Kash *et al.* [12] demonstrate that in a private community named DIME, after a period of freeleech, there are more download activities in the community. This is consistent with our theoretical result that during a pre-crunch state, injecting credit will increase the system throughput. While these papers mainly perform measurement-based studies, our paper is based on measurement, theoretical model, as well as extensive simulations. Further, we propose new strategies that effectively alleviate SRE's negative effects.

## 10 CONCLUSION

In this paper we have studied the effects of credit-based and SRE-based incentive policies employed in private P2P communities, from both the system-level and the user-level performance perspective.

Based on two user behaviors abstracted from real world observations, i.e., lazy-seeding and over-seeding, we examine the system-level credit dynamics and show that crunches and crashes can easily happen in private communities. We apply a theoretical analysis to characterize the conditions that lead to these extreme outcomes, based on which we design an *adaptive credit intervention mechanism* that proactively stops the system from seizing by temporarily changing the credit policies. A system that is predicted to crunch allows freeleech, and conversely, a system that is predicted to crash imposes freeseed. Simulation results show that our mechanism is very effective in avoiding crunches and crashes.

Given a private community that is sustainable, we further analyze its user-level performance. Our simulation results show that under SRE, swarms tend to be extremely oversupplied. Although achieving an increase

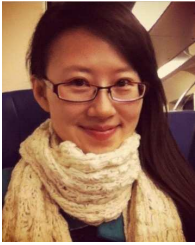
in the average downloading speed, the oversupply induces undesired effects including low upload capacity utilizations, extremely long seeding times, and an unfair playing field for late entrants into swarms. We propose four strategies to alleviate these problems. Particularly, *SRE with supply-based price* maintains a system-wide high downloading speed, achieves stable high upload capacity utilizations, and reduces seeding durations by three orders of magnitude as compared to the original SRE. When then the adaptive intervention mechanism is run in the background to check the extreme conditions for crunches and crashed, the system is ensured to have a high and sustainable performance.

## ACKNOWLEDGEMENTS

This work was supported by the European Union under grant agreement number 231200/QLectives Project.

## REFERENCES

- [1] <http://hdchina.org/>.
- [2] <http://chddbts.org/>.
- [3] <http://www.bitsoup.org/>.
- [4] M. Albert, *Parecon: Life After Capitalism*, London, 2003.
- [5] N. Andrade, E. Santos-Neto, F. Brasileiro, and M. Ripeanu, "Resource demand and supply in BitTorrent content-sharing communities," *Computer Networks*, vol. 53, no. 4, pp. 515-527, 2008.
- [6] D. Besanko, R. Braeutigam, R.R. Braeutigam, and J. Michael, *Microeconomics*, Wiley, 2010.
- [7] X. Chen and X. Chu, "Measurements, Analysis and Modeling of Private Trackers," *Proc. IEEE P2P*, 2010.
- [8] X. Chen, X.W. Chu, and Z. Li, "Improving Sustainability of Private P2P Communities," *Proc. IEEE ICCCN*, 2011.
- [9] M. Friedman, "Capitalism and Freedom: Fortieth Anniversary Edition," University of Chicago Press, pp. 192-194, 2002.
- [10] A.L. Jia, R. Rahman, T. Vinko, J.A. Pouwelse, and D.H.J. Epema, "Fast download but eternal seeding: the reward and punishment of sharing ratio enforcement," *Proc. IEEE P2P*, 2011.
- [11] I.A. Kash, E.J. Friedman, and J.Y. Halpern, "Optimizing scrip systems: Efficiency, crashes, hoarders, and altruists," *Proc. ACM Conference on Electronic Commerce*, 2007.
- [12] I.A. Kash, J.K. Lai, H. Zhang, and A. Zohar, "Economics of BitTorrent Communities," *Proc. NetEcon*, 2011.
- [13] G. Kaufman, "Banking and currency crises and systemic risk: A taxonomy and review," *Financial Markets, Institutions and Instruments*, vol. 9, 2000.
- [14] D. Lewis, *Louder voices: The corporate welfare bums*, 1972.
- [15] Z. Liu, P. Dhungel, D. Wu, C. Zhang, and K.W. Ross, "Understanding and Improving Incentives in Private P2P Communities," *Proc. ICDCS*, 2010.
- [16] 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," *Proc. IPTPS*, 2010.
- [17] D. Qiu and R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks," *Proc. Sigcomm*, 2004.
- [18] R. Rahman, D. Hales, T. Vinko, J.A. Pouwelse, and H.J. Sips, "No more crash or crunch: Sustainable credit dynamics in a P2P community," *Proc. International Conference on High Performance Computing and Simulation (HPCS'10)*, 2010.
- [19] G. Soros, "The worst market crisis in 60 years," *Financial Times*, January 22, 2008.
- [20] 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*, 2003.
- [21] C. Zhang, P. Dhungel, Di Wu, Z. Liu, and K.W. Ross, "BitTorrent Darknets," *Proc. IEEE Infocom*, 2010.



**Adele L. Jia** received her Mphil degree in 2009 from the Information Engineering Department of the Chinese University of Hong Kong. She is currently following a doctoral track in Computer Science within the Parallel and Distributed Systems Group, Delft University of Technology (TUDelft), the Netherlands. Her research interests are in the areas of incentive policies in peer-to-peer systems, user interactions in online networks, network economics, etc.



**Tamás Vinkó** received his PhD in 2007 from the University of Szeged, Hungary. He was a research fellow at the European Space Agency and postdoc researcher at the Delft University of Technology, the Netherlands. He is currently assistant professor at the University of Szeged, Hungary. His main research interest lies in mathematical modeling and optimization of large-scale distributed systems.



**Rameez Rahman** is currently serving as an Assistant Professor at Information Technology University, Lahore. He completed his PhD in 'Parallel and Distributed Systems' from Delft University of Technology, the Netherlands in 2011. He has invested his research efforts in utilizing inter-disciplinary ideas for building new, socially intelligent ICT systems. Such works have not only led to the development of more robust, efficient and fair protocols (results which have been published in many prestigious venues), but

can also open up vistas for social scientists to experimentally and empirically test the feasibility of various social principles, using ICT systems. Rameez envisions that much of the research impetus in the near future, in the sciences in general, and ICT in particular, is going to come from developing countries such as Pakistan. Towards that end, he intends to play his small part in fostering a research culture in Pakistan that can facilitate the realization of this vision.



**Johan A. Pouwelse** is an assistant professor at Delft University of Technology, specialized in Peer-to-Peer file sharing. He leads the P2P research team of a dozen people which created the Tribler P2P system. The Tribler group is the largest experimental research group in the field of P2P and responsible for several world-first innovations. With over one million downloads Tribler serves as a living laboratory and proving ground for next-generation P2P technology. Dr. Pouwelse is scientific director of P2P-Next and technical leader of QLectives, EU projects with a combined research budget of 26 million Euro. Previously Dr. Pouwelse delivered a statement for the FTC in Washington, was a visiting scientist at MIT, and spent several summers at Harvard to study mechanisms for cooperation.



**Dick H. J. Epema** received the MSc and PhD degrees in mathematics from Leiden University, Leiden, the Netherlands, in 1979 and 1983, respectively. Since 1984, he has been with the Department of Computer Science of Delft University of Technology, where he is currently an associate professor in the Parallel and Distributed Systems Group. Since 2011, he is also a part-time full professor of Decentralized Distributed Systems at Eindhoven University of Technology. During 1987-1988, the fall of 1991, and the summer of 1998, he was a visiting scientist at the IBM T.J. Watson Research Center in New York. In the fall of 1992, he was a visiting professor at the Catholic University of Leuven, Belgium, and in the fall of 2009 he spent a sabbatical at UCSB. His research interests are in the areas of performance analysis, distributed systems, peer-to-peer systems, and grids. He has coauthored more than 90 papers in peer-reviewed conferences and journals, was a general cochair of Euro-Par 2009 and IEEE P2P 2010, and the general chair of HPDC 2012.