2009 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises

BitTorrent or BitCrunch: Evidence of a credit squeeze in BitTorrent?

David Hales, Rameez Rahman, Boxun Zhang, Michel Meulpolder, Johan Pouwelse
Tribler Team
Technical University of Delft, Netherlands
Email: dave@davidhales.com, {rrameez, zhangboxun, michel.meulpolder, peer2peer}@gmail.com

***Abstract*—BitTorrent is a highly popular peer-to-peer file-sharing protocol. Much BitTorrent activity takes place within private virtual communities called "Private Trackers" - a server that allows only community members to share files. Many private trackers implement "ratio enforcement" where the tracker monitors the upload and download behaviour of peers. If a peer downloads substantially more than it uploads then service is terminated. Tracker policies related to credit effect the performance of the community as a whole. We identify the possibility of a "credit squeeze" in which performance is reduced due to lack of credit for some peers. We consider statistics from a popular private tracker and results from a simple model (called "BitCrunch").**

I. Introduction

BitTorrent (BT) is a highly successful peer-to-peer (P2P) file sharing protocol [4]. Originally BT was envisaged as an open protocol in which any peer could participate to cooperatively download files. To share a file a user needs to create a small .torrent file which uniquely identifies the file and binds it to a BT Tracker - a centralised server that keeps track of all the peers interested in a particular file. The .torrent file can then be distributed by any means, such as placing it on a webserver or e-mailing to a list. A user interested in the file can download the .torrent and activate their BT client software. The BT client will contact the tracker and connect to other peers interested in the file. The peers then cooperate to download the file by sharing pieces between them. A set of peers sharing a particular file is termed a "swarm".

A key feature of the BT protocol is the use of a "Tit-For-Tat" (TFT) strategy to control freeriding [2]. Put crudely, the idea of TFT is to incentivise peers to upload as well as download within a single swarm. Essentially, BT clients will stop uploading (sharing content) with those others who do not reciprocate.

However, although TFT provides incentives for those downloading from a given swarm to upload (to get more download) it does not provide incentives for two crucial activities: 1) for a peer to continue to share the file after it has downloaded the entire file - this is termed "seeding"; 2) for a peer to share a file in the first instance. In addition if a peer uploads more than it downloads in a swarm it cannot carry over any "credit" to a new swarm.

Recently, there has been a growth of "private tracker" based methods that attempt to provide incentives for these key functions by maintaining centralized accounts, that record upload / download behaviour of peers and apportion "credit" scores, and shutting out users who do not provide reasonable ratio to the system over some time period. We describe this approach in section II.

We examine statistics from a popular private tracker used to share TV shows in section III[1]. It is evident that such approaches *do* appear to incentivise seeding since we find that there are a huge number of seeders within the system. However, we find that a small minority of peers obtain a large amount of credit in the system. Similar results have also been obtained for another private BT community [1]. Such peers, who hog the majority of the credits in the system, have been defined as "hoarders" in the literature [8]. Hoarders have a negative impact on the system because their presence effectively means that there is less credit in circulation leading to a credit squeeze. As a result other peers suffer, unable to obtain desired services due to being short on credit. This creates a credit squeeze. *We define a credit squeeze as a situation in which, due to lack of credit, the efficiency of the system is significantly reduced.*

Using a simplified simulation model we show that even when all major factors are equal, a skewed distribution of credit in the system emerges and leads to a credit squeeze reducing system efficiency. Interestingly, we show that *adding capacity to the system can, counter-intuitively, reduce performance* due to shortage of credit. We also examine a simple method used by the measured private tracker that ameliorates the credit squeeze in our model. The BitCrunch model description, simulation results and discussion are given in section IV, V and VI.

This paper reports preliminary results in what we believe is an under explored research area. Our aim is to open a new line of work, within BitTorrent studies, which examines the role and function of macro-economic policies at community level in order to increase community efficiency.

II. Private trackers

Private trackers require users to registered before they are allowed to download .torrent files and participate in swarms.

A major function of private trackers is to implement "ratio enforcement". This means the tracker monitors how much

[1] We prefer to keep details of the tracker anonymous but all data is available on request of the authors.

1524-4547/09 $25.00 © 2009 IEEE
DOI 10.1109/WETICE.2009.22

upload and download bandwidth is used by each peer over time. If a peer downloads more than it uploads (a form of freeriding) then the download service is terminated. Ratio is persistent over peer sessions and is stored centrally with the tracker. This provides a mechanism for peers to build-up a positive "ratio" or earn credit in the system over many sessions and swarms. This means that peers who seed content to others earn credit that they can "cash in" at a future time and / or in a different swarm.

The tracker acts as an accounting system in which the balance is constant over time. When peer i uploads to peer j then i's account is credited and j's account is debited by the amount exchanged (some number of megabytes say). Only those peers with positive credit may continue to download from others.

If we assume that the tracker performs ratio enforcement by terminating any downloading by peers with zero credit then the total credit in the community will always be a constant value C but its distribution among peers will change over time to reflect the actual exchanges that have occurred. Hence a peer that seeds a very popular file to many others and has a high physical upload rate would expect to accumulate a high proportion of the available credit whereas those that have downloaded much but uploaded little will expect to have small or zero share of the credit.

Different private trackers run variants of this form of policy. They may allow negative credit or new artificial credit for new. Also, as we be seen, some trackers allow new credit to be created in the system overtime.

III. Evidence of a credit squeeze?

Given peers with different file preferences and upload / download rates the result of such a process often leads to a highly skewed distribution of credit over all peers. In fact we have found that a small "rich club" of peers appear to hold a large proportion of credit in a real private tracker community.

A. Statistics from a private tracker

Table I shows 7 days of statistics gathered from a popular private tracker (over the period 06/02/09 to 12/02/09). This tracker has a reported daily population of approximately 50,000 peers serving of the order of 10,000 .torrents. Peers earn credit by seeding and spend it by leeching (downloading). It is possible for peers to earn credit by uploading while downloading, through the BitTorrent TFT process, but we found that there is strong evidence that most swarms are over-seeded and hence we discount this as a major factor in credit dynamics the tracker we measured. In over-seeded swarms downloaders do not have to trade with downloaders but can download directly from seeders.

The tracker we measured rewards seeding by providing credit to seeders at the "bonus rate" of 1.5 times upload. Hence a peer uploading one byte will receive 1.5 credits while downloading one byte will cost 1 credit. This means the amount of credit in the system is always increasing.

TABLE I
STATISTICS FROM A POPULAR PRIVATE TRACKER OVER 7 DAYS

Day	T	Δ	Δ_0	δ	S/L
1	48	24	17	0.23	26
2	40	20	15	0.25	26
3	50	25	12	0.16	25
4	67	33.5	17	0.17	25
5	52	26	19	0.24	25
6	46	23	15	0.21	25
7	87	43.5	17	0.13	25
Ave.	56	28	16	0.19	25

Since we have access to statistics detailing total throughput estimates and the actual credit of the top 10% of peers (top 5000 by credit) we can calculate an estimate of the amount of new credit accumulated by the top peers as a proportion of all new credit created over time.

Statistical data is collected by scraping the web pages of the tracker. We remove all invalid data due to server failures from our analysis. Because the amount of invalid data is small, it does not affect our analysis.

T shows the total throughput of the system. Since we assume the system is closed, it holds that the total throughput is equal to both the total upload (U) and total download (D) in the system. Δ is the total credit increase in the system per day. The total credit increase comes solely from the *credit bonus* of 0.5 rewarded to uploading. Therefore, $\Delta = \frac{1}{2}U$.

Δ_0 is the increase in total credit of the top 10% of the population (in Terabytes). δ represents the minimum fraction of the total credit that goes to this top 10% (We explain how we arrive at this lower bound in the next section). S/L shows the ratio of seeding to leeching sessions over the entire set of swarms served by the tracker. We also calculated the turnover of the top peers which indicates how many of the top 10% of peers, by credit, change each day. We found this to be minimal averaging 0.2% over each day (not shown in the table).

Note we can see that the ratio of seeding to leeching sessions (S/L) is high. This means that many peers are seeding much content - presumably to earn credit. Yet the top 10% of peers take a large proportion of the new credit created in the system (δ). We conjecture that this level of credit increase in the top peers is a result of high upload bandwidth and the high S/L is a result of credit starved peers seeding many swarms. Hence this evidence is consistent with the notion of a *credit squeeze* but not directly indicated by it since we do not know that adding more credit would improve throughput.

B. Lower bound for top peers

We will now analyze the minimum fraction of Δ that is obtained by the top 10% of peers. Let Δ_0 be the credit increase of only these peers. Let U_0 and D_0 be the total number of Terabytes uploaded and downloaded, respectively, by these peers. The credit increase of these peers can then be expressed as:

$$\frac{3}{2}U_0 - D_0 = \Delta_0 \tag{1}$$

Note that it can't be known how much of this credit increase is part of the new credit injected into the system and how much has been taken away from peers outside the top 10%. However, we will show that a certain *minimum fraction* of new credit injected into the system is gathered by the top peers. Since $D_0 \geq 0$ it follows that $\Delta_0 \leq \frac{3}{2}U_0$. The new credit generated or credit bonus received is $\frac{1}{2}U_0$. The credit bonus is therefore at least $\frac{1}{3}\Delta_0$. For the fraction γ of the total credit bonus that is obtained by only the top 10% of peers, it then holds that:

$$\gamma = \frac{\frac{1}{2}U_0}{\Delta} \geq \frac{\frac{1}{3}\Delta_0}{\Delta} \tag{2}$$

We can trivially derive the upper bound of γ. Since naturally $U_0 \leq U$, it holds for the credit bonus that $\frac{1}{2}U_0 \leq \frac{1}{2}U = \Delta$. Therefore, $\gamma \leq 1$. This corresponds to the case where the peers outside the top 5000 have uploaded nothing whatsoever. Furthermore, since $D_0 \leq D$ and $D = U$ it holds as well that $D_0 \leq 2\Delta$.

In our measurements, $U = 56$ and $\Delta_0 = 16$ on average. Therefore, the top 10% peers obtain on average at least a fraction of 0.19 of the total credits injected in the system. As the top 10% of peers are only a small fraction of the total community, the gap between 'rich peers' and 'poor peers' is inevitably growing bigger.

IV. BitCrunch model description

In order to explore the minimal conditions under which certain credit dynamics occur, we have designed a model (BitCrunch) containing the essential properties of credit systems. We stripped away the complexities of real communities, so that the underlying forces of credit become clear and can be analysed.

A. Peers

The community is represented by a set of peers ($\mathcal{P}$). Each peer i has a predefined and fixed upload (up_i) and download ($down_i$) capacity (in units of data per unit of time).

Here we use a highly simplified user / client model. Peers are online at all times. At any given time a peer is seeding some number of swarms (S) and downloading from some number of other swarms (D). When a peer has finished downloading a file it moves it from it's download list to its seeding list. Peers seed files for some predefined and fixed amount of time and then remove them from their seeding list.

If the number of currently downloading files is less than D then the peer selects new swarms to download until D number of files are being downloaded. If adding a new seed to the seeding list causes the size to exceed S then the oldest seeding file is removed from the list. In this way each peer will always be downloading from D swarms and seeding a maximum of S swarms.

In our initial experiments we use a minimal form of this scheme by setting $D = S = 1$ and the maximum seeding time set to infinity. This means that each peer is always downloading in one swarm and seeding in one other swarm.

B. Swarm capacity

The tracker supports a set of swarms ($\mathcal{S}$). Each swarm contains some number of seeder and leecher[2] peers (including possibly zero). Each seeder holds a copy of the entire file which the swarm is distributing. Each leecher contains some proportion of the file. Since every peer is assigned an upload and download rate we can calculate the total demand (sum of all leecher download rates) and the total supply (sum of all seeder upload rates). Hence for a given swarm S_i:

$$supply(S_i) = \sum_{j \in S_i} up_j$$

$$demand(S_i) = \sum_{k \in S_i} down_k$$

where j is a seeder peer in S_i with up_j upload capacity and k is a leecher peer in S_i with $down_k$ download capacity.

In order to create transactions based on supply and demand we adopt a highly simplified swarm model. If supply matches demand then all leeching peers receive their entire download capacity and all seeders use their entire upload capacity. If demand exceeds supply (under supply) then all seeders use their entire upload capacity but each leecher k only receives:

$$download(k) = \frac{supply(S_i)}{demand(S_i)} * down_k$$

Hence if demand was twice supply then each leecher would only receive half of its download capacity. Conversely if supply exceeds demand then each leecher receives its entire download capacity but each seeder j uploads only:

$$upload(j) = \frac{demand(S_i)}{supply(S_i)} * up_j$$

Again this means that if supply was, say, twice the demand then each seeder would use only half its upload capacity.

As will be seen later our initial simulation experiments assume all peers have equal upload and download capacities meaning all peers in the same swarm obtain identical service.

C. Ratio enforcement

All peers begin with an equal amount of credit which for all peers in the system sums to C. When peers download, their individual credit score is reduced accordingly. Conversely, uploaders have their credit score increased. In this way C stays constant over time but the distribution, over peers, changes depending on download and upload behaviour.

If a peer runs out of credit it can no longer download and is excluded from the list of active downloaders in any swarm it is trying to download from. When a peer enters this state it is considered "broke" and cannot download again until it earns credit from its seeding activities in other swarms.

[2] We use the term leecher and downloader synonymously in this paper.

V. Simulation experiments

To examine the relationship between initial credit, efficiency, and credit dynamics in our model we produced two sets of simulation runs. In the first set of "baseline" runs we initialised each peer with equal upload and download capacities and credit. No new credit was created over time. In the second set of "unequal capacity" runs we added capacity to the system by setting a minority of peers to have higher upload capacity than the majority and, in addition, we experimented with creating new credit in the system by awarding a "seeding bonus".

The simulation runs proceed in discrete time units (or cycles). One time unit involves each swarm transacting uploads and downloads based on the supply / demand model previously described.

A. Baseline runs

We started each run with all peers equally sharing the initial credit C and having equal upload and download capacities. All swarms were assigned equal popularity and each file had the same size. We defined efficiency as how much data was exchanged in units (throughput) over a fixed time period.

We set all file sizes to 10 units. We set the upload and download capacities of each peer to 1 unit (per time unit). We performed runs for three different initial credit values (1, 10 and 100 per peer). We set the number of peers to 500 and the number of swarms to 100. This ensures sufficient peers to create meaningful levels of supply and demand in each swarm.

These values give a highly balanced baseline in which all things are equal for each peer. In the real world of private trackers such balance would not occur since peers have different upload / download capacities, availability and user behaviour. In addition swarms follow non-equal popularity distributions [1]. We performed these simulations to determine if credit squeeze phenomena can be observed in our simple model where key parameters are held equal.

Table II shows results from simulation runs with different initial credit amounts (peer per) given by C. Each value represents an average of 10 simulation runs to 2000 cycles (variance was negligible). T is the total cumulative throughput at the end of the runs. This is shown as a proportion of the maximum throughput that could be achieved if no peer is allowed to become "broke" - i.e. if all peers have infinite credit. As can be seen T increases as C increases. β gives the proportion of peers with zero credit (broke peers) at the end of each cycle averaged over the last 10,000 time units. G gives the *Gini inequality measure*[3] for peer credit at the end of each cycle - again averaged over the last half of the runs. φ gives the turnover of the top 10% of peers (by credit) as a proportion. This is calculated after every 100 time units and is averaged over the last half of the runs. Hence turnover (φ) gives a measure of the "credit mobility" of peers as a proportion of change in the top 10% of peers over time.

[3]The Gini coefficient [0..1] characterises inequality with 1 being the most unequal (one peer holds all credit) and 0 being complete equality.

TABLE II
Results from baseline simulation runs

C	T	β	G	φ
1	0.58	0.36	0.87	0.84
10	0.81	0.20	0.77	0.43
100	0.97	0.06	0.59	0.10

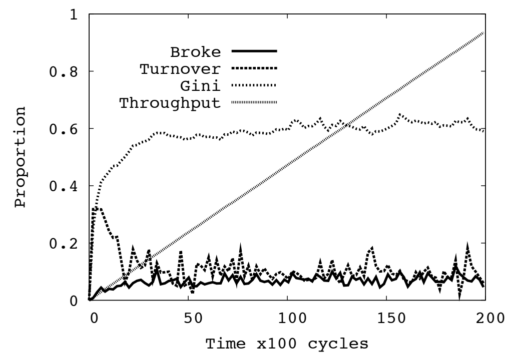

Fig. 1. Typical time series of a single simulation run of the model where initial credit is set to 100 units per peer with peers having equal capacity.

Given our definition of a credit squeeze it is evident that when initial credit (C) is low then throughput (T) is also low. However, when C = 100 units of credit per peer then T is 94% of the maximum T achievable. Notice also that peer credit inequality (G) decreases as C increases but that even when C = 100 there is still high inequality with a Gini value of 0.6. Turnover (φ) also decreases as credit increases. However, even when C = 100, turn-over is still high implying that the top 10% of peers by credit could change completely within 1000 time units.

Figure 1 shows a typical time series from a single simulation run for C=100. Note that, after an initial period, values settle within stable bounds. Because the proportion of broke peers (β) is low, due to high initial credit, the total cumulative throughput (T) almost reaches the maximum throughput found when credit was infinite.

B. Unequal upload capacities

In order to examine the credit dynamics when some peers have differing upload and download capacities we performed a further set of simulation runs in which 10% of peers were intialised with upload and download capacities of 10 units. The other 90% were given a download capacity of 10 units but an upload capacity of only 1 unit. Here we capture the notion that only a small number of peers have high upload capacity whereas all peers have high download capacity. We kept all other parameters the same as in the baseline runs.

Table III shows the results obtained. As could be expected we see a high level of credit squeeze for the fixed credit

TABLE III
RESULTS FROM UNEQUAL UPLOAD CAPACITIES SIMULATION RUNS

C	T	β	G	φ
1	0.56	0.39	0.90	0.82
10	0.71	0.32	0.93	0.44
100	0.77	0.29	0.94	0.60
100++	0.97	0.01	0.71	0.00

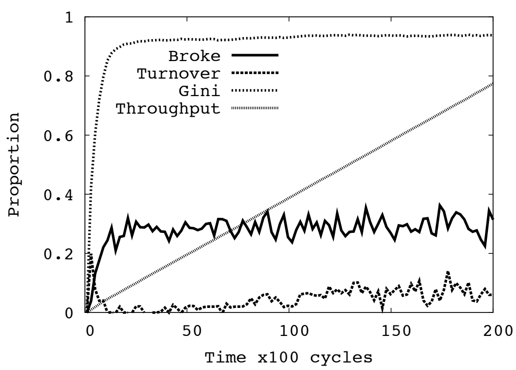

Fig. 2. Typical time series of a single simulation run of the model where initial credit is set to 100 units per peer with peers having unequal capacities.

runs (C=1, 10 and 100). Notice however that for the C = 100++ runs we have a minimal credit squeeze effect. In the C = 100++ runs we initialised each peer with C = 100 but implemented a "seeding bonus" scheme in which seeders receive an additional 50% credit bonus on all upload. This means a seeder uploading 1 unit receives 1.5 units of credit.

Interestingly we found that the throughput value, T, was actually lower in the fixed credit scenarios[4] even though substantial capacity had been added to the system (in upload and download).

Figure 2 shows a typical time series from a single simulation run for C=100. Note that after an initial period values settle within stable bounds. Notice that the proportion of broke peers (β) is much higher than the previous baseline time series and hence the cumulative throughput (T) of the system is reduced.

VI. DISCUSSION

From the empirical results in section III we conjectured evidence of a credit squeeze. Our simulations results show that *even in a trivial model where all peers have the same capacities and user behaviour, all swarms have equal popularity and all peers start with equal credits, the performance of the system may be inhibited by credit shortages*. This is because high levels of credit skew emerge due to the fact that a peer can only upload a file it has already downloaded.

[4]By this we mean that even if T is expressed as an absolute value, rather than as a proportion of maximum capacity, it is lower than in the baseline runs

We also observe that in such scenarios *adding extra capacity to the system, in the form of upload and download, can actually reduce the performance*. This is highly counter intuitive and something that should be avoided because it implies lack of scalability.

Finally, we found that *by injecting new credit into the system in the form of a "seeding bonus" a credit squeeze can be ameliorated* when peer capacities are unbalanced.

The results from our empirical evidence and model must be qualified because we have excluded credit earned during TFT behaviour - where downloading peers exchange data with other downloaders. Based on our private tracker measurements we have concluded that the majority of exchange is between seeders and leechers but we cannot be sure of this.

Additionally our assumption that swarms resolve supply and demand in an equitable way is not the case in BitTorrent due to limited upload "slots" and the randomised nature of peer selection over time. Our abstraction is that over time and swarms this might be comparable to equal shares. But again we cannot be sure of this.

In our model we observed that our swarms became quickly skewed, containing many leechers and a small number of seeders and vice versa. Why does this happen when selection of swarms is uniformly random? This is because any initially small imbalance becomes exaggerated through positive feedback. A swarm with high leecher / seeder ratio gets slow and hence tends to "recruit" more leechers. A swarm with a low ratio will tend to turn new leechers into seeders very quickly, creating more seeders. But this would not happen in BitTorrent because of TFT effects (which our model excludes). Although, coincidentally, the flash-crowd swam life-cycles observed in real systems could produce similar skews [1].

Since we have not put freeriders into our model we have not explored the tradeoff between amount of credit in the system and de-incentivising freeriding. A freerider would take as much credit from the system as they could without reciprocating. This could be a interesting area for future research.

VII. RELATED WORK

BitTorrent is a relatively new area of study. However, despite its recency, a great amount of work has been done on studying the BitTorrent protocol and communities. Many research studies have been conducted to determine the robustness, scalability and performance of BitTorrent-like systems [15], [14], [9], [11], [10], [7], [6], [3].

It should be noted that these works focus on the single swarm while we are concerned with the multiple swarm scenario. Other works which have focused on multiple swarms address the so-called 'seeder promotion problem'. The basic claim across these papers is that while the TFT policy of the BitTorrent protocol could be considered practical for single swarms, it doesn't offer incentives for peers to seed content after finishing their own downloads [12], [13], [5]. All these works include a kind of monetary mechanism to incentivize cooperation among peers.

A recent work by Andrade et al. [1] examines the supply and demand for resources in public and private BitTorrent file sharing communities. Their main contributions include showing that only a minority contributes the majority of the resources and that the upload contribution of peers is not correlated to the time that they invest in seeding content. These findings lend credence to our hypothesis that a minority of the users hogs the majority of the credits while a large majority are unable to earn credit despite seeding content for long durations.

Credit crunches and crashes have been studied in Scrip Systems by Kash et al [8]. They show that in a P2P system, both an overabundance of money supply and its shortage lead to inefficiency. An overabundance in the money supply leads to a monetary crash where no one is willing to work and freeriding is encouraged. On the other hand, a shortage in the money supply leads to peers going broke and not being able to afford services in the system. This work is different from ours in that we place our analysis firmly in the very practical domain of private BitTorrent communities and bring forth real life measurements to support our hypothesis.

VIII. Conclusion

We have introduced the notion of a "credit squeeze" where lack of credit impacts the efficiency of private tracker systems. We presented some initial statistics from a popular private tracker that are consistent with this idea but are not conclusive.

We defined a simplified model of a private tracker and found that insufficient initial credit can lead to a credit squeeze even when all peers have equal capacities, user behaviour, and all swarms in the system have equal popularity. Through simulation we showed that in such systems highly skewed credit distributions emerge between peers but that over time peers have high mobility of credit rank. This means the credit rich do not stay rich and poor do not stay poor but at any given time there are rich and poor.

We also performed simulations where a minority of peers had much higher upload bandwidth capacity than other peers. Here we found that the minority became highest in the credit rank, as to be expected, but the imbalance led to high system level inefficacy due to a credit squeeze in the lower capacity peers. Hence adding capacity to the system actually reduced efficiency because the rich minority, over time, accumulate the majority of credit in the system. Interestingly, we found that by applying a policy modeled on that used by the private tracker, from which we collected statistics, solved the credit squeeze at the expense of reduced rank mobility (it went to zero). This policy involved giving seeders a "bonus credit" of 50% of their upload. Essentially this equates to imposing ratio of at least 2/3 on all peers (i.e. they only need to upload two thirds of what they download to stay in credit).

The work presented in this paper is at an early stage. We have a hypothesis that a lack of credit circulation, due to a minority of peers obtaining the majority of the credits, can severely degrade the efficiency of private trackers. However, while we do have evidence from a single private tracker that the gap between the rich peers and slow peers is widening, we did not have access to the relevant data that could prove that this phenomenon also hampers the efficiency of such systems.

However, using a simplified model of a private tracker, we were able to show through simulation that the disparity of wealth (credits) among the peers leads to a credit squeeze. In the model, we exclude many phenomena that could lead to credit skews such as: injecting of new content; differing swarm popularity and swarm "life-cycles"; peer availability and differing user behavior (including freeriding). We believe all of these factors are highly important in shaping how private trackers actually perform.

References

[1] Andrade, N. and Santos-Neto, E. and Brasileiro, F. and Ripeanu, M. Resource demand and supply in BitTorrent content-sharing communities. In *Computer Networks*, 2008

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

[3] Bellissimo, A. and Levine, B.N. and Shenoy, P. Exploring the use of BitTorrent as the basis for a large trace repository. *University of Massachusetts Technical Report*, 2004

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

[5] Garbacki, P. and Epema, D.H.J. and van Steen, M. An Amortized Tit-For-Tat Protocol for Exchanging Bandwidth instead of Content in P2P Networks. In *Proc. First International Conference on Self-Adaptive and Self-Organizing Systems*, 2007

[6] Izal, M. and Urvoy-Keller, G. and Biersack, E.W. and Felber, P.A. and Hamra, AA and Garces-Erice, L. Dissecting BitTorrent: Five Months in a Torrent's Lifetime. In *Passive and Active Network Measurement*, 2004.

[7] Jun, S. and Ahamad, M. Incentives in BitTorrent induce free riding. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, 2005

[8] Kash, I.A. and Friedman, E.J. and Halpern, J.Y. Optimizing scrip systems: Efficiency, crashes, hoarders, and altruists. In *Proceedings of the 8th ACM conference on Electronic commerce*, 2007

[9] Legout, A. and Urvoy-Keller, *G.* and Michiardi, P. Rarest first and choke algorithms are enough. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006

[10] Locher, T. and Moor, P. and Schmid, S. and Wattenhofer, R. Free Riding in BitTorrent is Cheap. In *5th Workshop on Hot Topics in Networks (HotNets)*, 2006

[11] Piatek, M. and Isdal, T. and Anderson, T. and Krishnamurthy, A. and Venkataramani, A. Do incentives build robustness in BitTorrent. In *Proc. of NSDI*, 2007

[12] Ramachandran, A. and Sarma, A.D. and Feamster, N. BitStore: An Incentive-Compatible Solution for Blocked Downloads in BitTorrent. In *2nd Joint Workshop on Economics of NetworkedSystems and Incentive-Based Computing*, 2007

[13] Sirivianos, M. and Park, J.H. and Yang, X. and Jarecki, S. Dandelion: Cooperative Content Distribution with Robust Incentives. In *NetEcon*, 2006

[14] Stutzbach, D. and Zappala, D. and Rejaie, R. The scalability of swarming peer-to-peer content delivery. In *Proceedings of Networking*, 2005.

[15] Urvoy-Keller, *G.* and Michiardi, P. Impact of Inner Parameters and Overlay Structure on the Performance of BitTorrent. In *Proceedings of INFOCOM 2006. 25th IEEE International Conference on Computer Communications*, 2006