

On the sustainability of credit-based P2P communities

Tamás Vinkó · Helga Najzer

Received: date / Accepted: date

Abstract Evidences show that besides their incredible performance, private BitTorrent communities which employ credit-based contribution incentives can also confront with system seize-up. Although the sustainability of these communities can be established in a wide range of system parameters, it is still not fully understood under what initial conditions can this be assured. In this paper, with the usage of extensive simulation results, we investigate the effects of the file size and the impact of initial credit amount distribution in order to deduce further conditions of sustainability.

Keywords BitTorrent community · credit · incentive · sustainability · simulation

1 Introduction

Private BitTorrent communities are content sharing systems using peer-to-peer technology, in which users are incentivised to stay online and provide uploading service after finishing the downloading of the files. This is not provided in the standard BitTorrent protocol [6]. These incentive mechanisms, for example, are the sharing ratio enforcement (SRE) or credit system. In this paper we study credit-based BitTorrent communities. In this kind of system a registered user can obtain a particular file of interest in exchange of some virtual credit. Upon registration, which is usually invitation based, each peer is provided with some initial amount of credit. A file can only be requested if the user has enough credit to download it. Credit collection can be done with uploading.

Recent papers have shown that, similarly to a general financial market [15] or scrip system [13], private BitTorrent communities could also face with credit instability [8, 10, 19]. Basically, there are three statuses: *crash* (in case there are no uploaders

T. Vinkó, H. Najzer
University of Szeged, Institute of Informatics
H-6720 Szeged, Árpád tér 2, Hungary
Tel.: +36-62-546 193
Fax: +36-62-546 397
E-mail: tvinko@inf.u-szeged.hu

due to too much credit distributed evenly), *crunch* (there are no downloaders due to lack of credit) or *sustain*. Note that sustained status is usually temporary, which could be held for both short term or longer term. On the other hand, crash and crunch are both definitive, they can only be fixed with external interventions [9, 19].

Our goal here is to revisit the problem of sustainability of BitTorrent based credit systems in order to derive further conditions by means of extensive simulations. We investigate initial conditions regarding the credit amount and credit distribution among the rich peers (i.e. those who have enough credit to start downloading a file). We also show that certain ratios of rich peers could lead to unstable system in case of larger file size.

2 Definitions

Content sharing over a large scale network such as the Internet is usually done with the usage of peer-to-peer (P2P) technology. In contrast to centralized solutions, P2P networks are decentralized by nature. The most famous and widely spread P2P protocol is BitTorrent [6]. Its extreme scalability and robustness is provided by its concept about how the sharing of the content (usually media files) is handled. In the BitTorrent protocol there are two types of users. *Seeders* are those who have the complete copy of the file(s). In case these type of users are online, they provide download services to *leechers*, i.e. to those who want to obtain files. Together with a given file, which is subject to be shared, the set of leechers and seeders are called a *swarm*. The associated file to a swarm is split into smaller *pieces* and the download is done in the following way. Leechers do not only get pieces from seeders, but they upload to and download from each other using rarest-first scheme. This implies that larger swarm usually guarantees faster download time.

It is easy to see that sharing files using any kind of decentralized P2P network needs to incorporate contribution incentives which encourage the participating peers to provide resources to the others. The robustness of BitTorrent, especially in case of high leechers/seeders ratio, is guaranteed by the built-in tit-for-tat (TFT) mechanism, inspired by the most robust solution to the famous experiment on the iterated Prisoner's dilemma game [2]. On the other hand, seeding is not at all incentivised. Thus a BitTorrent network could confront with large amount of users with hit-and-run behavior [22]. One possible solution to this problem is the so-called private BitTorrent communities, also called BitTorrent darknets [23].

In a private BitTorrent community users are required to register themselves at a central server called *tracker*. This tracker maintains not only the list of the available content, but also statistics about each users' download and upload amount. Its role is mainly to enforce peers to share. This is usually done by *sharing ratio enforcement* (SRE) or usage of virtual credit. In the former case peers need to keep their download/upload ratio above a certain level (usually 0.7) in order to be allowed to obtain new files, while in the latter, users can download new content only if they have enough credit. In this paper we study BitTorrent communities employing *credit based sharing incentive*. Under this kind of policy, each user is required to maintain

non-negative credit, which is calculated as the amount of data uploaded minus the amount of data downloaded by the given user.

Note that many private communities use SRE and credit (or point) mechanisms together [5]. An example of this kind of community is CHDBits, which is extensively studied in [12]. In this paper we assume that the considered BitTorrent community employs pure credit based policy. A real private community of this type is TVTorents.

3 Related work

Significant research effort has been dedicated to the study of private BitTorrent communities. Measurement studies show that compared to open/public communities they (i) usually provide their users with superior performance compared to open communities [17], (ii) differ with respect to torrent evolution, content distribution [4] and their users behaviors [5], and (iii) have different resource demand and supply [1]. At the same time, several drawbacks of these highly incentivized systems have been identified. Some of these disadvantages were reported in [16]. Counter-intuitively, long seeding time does not necessary lead to high sharing ratio (or to be rich in the terminology of credit systems). This fact was demonstrated and possible strategies were discussed in [11, 12]. Incentive schemes in peer-to-peer communities based on credit have been proposed in many papers, e.g. [7, 20, 21].

Our current paper is complementing the work done in [19]. In [9, 10] another kinds of extended considerations have been done, different from ours, mainly focusing on remedy schemes avoiding crashes or crunches. We did not aim at proposing further mechanisms for preventing systemic risk, as the already developed ones can be applied in the communities we have considered.

In [3] a model to identify the optimal stable seeder-to-leechers ratio (*SLR*) in private BitTorrent communities has been provided. It has been derived that, under certain assumptions, the *SLR* value should be in the range of [1.67, 1.73]. Our findings are in good alignment with this as we also find that the highest performance, measured by the system's average throughput, is reached when the initial proportion of rich peers (leechers) is set to be 60%, the system converges to near $SLR = 1.6$, in particular at larger file size, see Fig. 1a and 4a. While the results in [3] are built on a mathematical model, our work is based on extensive simulations which enables much in-depth analysis of the underlying systems parameters.

Increase in download activity as the aftermath of freeleech period has been demonstrated in [14]. The study is based on real-world measurement of the DIME private BitTorrent community. Our simulation results are in line with these, however, we point out that credit injection must be done with care as it can lead to performance drop in several cases depending on the actual composition of poor and rich peers.

4 System model

Basically, the system model we use in this paper is very similar to the one used in [19]. Here we give a short overview, detailing the modifications we applied.

The simulator is an abstract model of the real private BitTorrent communities as it includes the most important (high level) properties of them. It works based on *cycles*, where one cycle corresponds to one time unit. Each peer gets activated in each cycle and does one of the following activities: either it continues the already started download or upload, or it can initiate new upload or download. We assume credit-based system in which credit can be earned by uploading, and credit can be spent on downloading.

Tracker The tracker implemented in the simulator has the similar role as in a real system as it functions as a central component. It keeps track of those swarms which are available to the participating users. Moreover, it monitors the upload and download amount of each peer. It applies credit-based incentive mechanism: if a peer does not have enough credit to download a file, it has to collect some more with seeding.

At the start of a simulation run, the ratio of rich and poor peers can be set up. By default, rich peers get exactly that amount of credit which is enough to download one file, while the poor peers have zero credit. The poor peers are the seeders and rich peers are the leechers. Note that we simulate a closed community, thus the overall credit amount is a fixed constant.

Users The community is represented by its peers. The system has N peers online, each of them with upload capacity U , which basically defines the number of upload slots can be used for upload in a round. No new peers are arriving. This assumption apparently resembles reality, as it is usually not easy to get registered in private BitTorrent communities. No hit-and-run behavior is considered. Furthermore, it is assumed that each peer uses its full bandwidth capacity.

Swarms The community consists of S swarms. In reality the number of swarms a peer participates in is not limited. However, a peer's sharing ratio (or credit amount) is aggregated across the swarms, thus we can assume here that each peer participates only in one swarm at any given time.

Data flow In modelling the download procedure we follow the tit-for-tat (TFT) mechanism of BitTorrent: within a swarm the seeders are uploading pieces of the file to the leechers, and leechers can exchange pieces among each other. All users want to obtain all available files in the community. In case a peer has no enough credit, it stays in the current swarm (in which it downloaded the corresponding file) to seed until it becomes rich. Then the peer randomly selects a swarm to join as a leecher. In a real-life community the set of available contents are usually increasing (or at least changing). In the simulator we do not identify the swarms, which can be interpreted as an ever-changing pool of content.

Transactions Each file is divided into pieces of fixed size. Similar to the BitTorrent protocol, each peer has 4 uploading slots and thus is able to upload (maximum) 4 pieces per one cycle. Within a cycle all the peers get activated in random order. When a peer p is being active, it randomly selects 4 leechers in the swarm and uploads pieces to them following BitTorrent's rarest-first scheme, i.e. pieces which are less

replicated in the swarm have higher chance to be uploaded. In case the selected peer has all the pieces provided by the uploading leecher p , then p selects another leecher in random fashion. This mechanism guarantees that all uploading capacities are fully activated in each round.

Mathematical model In [10] a model for predicting crashes and crunches was proposed. The idea is to count the number of prosperous leechers and seeders who, given the current situation in the BitTorrent community, have the credit earning potential to be rich in the forthcoming. Based on this idea adaptive credit intervention mechanism was used to avoid crash or crunch. Here we revisit the model with the aim of giving a more elaborated version. Based on this slightly improved formalism further indicators about sustainability of the community using empirical investigations will be derived.

The following notation is used. The number of swarms in the community is S ; the number of leechers and seeders in swarm ℓ at time t is $x^\ell(t)$ and $y^\ell(t)$, respectively; x_i^ℓ is the leecher i in swarm ℓ and y_j^ℓ is seeder j in swarm ℓ ; $c_x^\ell(i, t)$ is the credit of leecher i in swarm ℓ and $c_y^\ell(j, t)$ is the credit of seeder j in swarm ℓ at time t ; $p_k^\ell(t)$ is the proportion of the file that leecher x_k^ℓ has at time t ; C is the amount of credit required to download a file (according to our assumption $C = F$). Moreover we will use η as the efficiency of piece exchange among leechers. According to [18] parameter η has the form

$$\eta = 1 - \left(\frac{\log F}{F} \right)^k,$$

where $k = \min\{x^\ell, K\}$ and K is the number of leechers to which a peer can connect, given by the protocol's definition.

Two sets are introduced, for which we use the main idea of the model in [10]. Our model here is more general as it is using the concept of η , i.e. the efficiency of piece exchange amount leechers. The set $X_\ell(t)$ contains those leechers who will be still rich after finishing the download of the file associated with swarm ℓ . Formally

$$X_\ell(t) = \{x_i^\ell : c_x^\ell(i, t) + p_i^\ell(t)C + \frac{\eta(x^\ell(t) - 1)}{y^\ell + \eta(x^\ell(t) - 1)}C \geq 2C\}.$$

The set $Y_\ell(t)$ consists of those seeders who have the credit earning potential to become rich. This can be formalized as

$$Y_\ell(t) = \{y_j^\ell : c_y^\ell(j, t) + \sum_{k=1}^{x^\ell(t)} \frac{1 - p_k^\ell}{y^\ell + \eta x^\ell(t)}C \geq C\}.$$

These sets will be quantitatively investigated in our simulations.

5 Simulation results

In order to understand the possible outcomes we performed several simulation runs using different initial parameters. In each run we used $N = 1,000$ online peers and

the number of swarms were $S = 100$. Every simulation run is done for 2,000 cycles and each run is independently executed for 5 times. By default, rich peers have the amount of initial credit which is enough to download one file, while poor peers (initial seeders) have zero initial credit. At the initialization phase peers are distributed uniformly at random among the swarms. This might result in swarms with zero seeders and some leechers. For each of these swarms, if any, a seeder from another swarm (which has more than 1 seeders) gets relocated, making each file available to download. The total amount of credit in the community is equal to $N \cdot F$ multiplied by the proportion of rich peers. The initial proportion of rich peers, denoted by R , is varied from 10% up to 90%. For a given community set up, the size of all the files are equal (F) and the size of a piece is 1. As it was shown in [19], a community crunches if $F = 10$ and $0.1 \leq R \leq 0.2$, while with $F = 10$ and $R \geq 0.9$ it crashes.

5.1 Effect of different file size

In the following we aim at investigating whether the size of the file in the system has any effect on the sustainability. In the simulations the size of the file is increased from 10 up to 100, and for each fixed file size we analyse the initial proportion of rich peers varied from 30% up to 80%. In case of $R = 0.3$ the community can only be sustainable if $F = 10$, thus we do not include the corresponding simulation results in the upcoming discussions.

5.1.1 Homogeneous peers, default case

In these simulations every peer has the same upload capacity, which is 4 pieces per round. Fig. 1 shows the average, minimum and maximum throughput in the system under different initial proportions of rich peers. It can be clearly noticed that larger file size leads to larger maximum throughput. However, from Fig. 1c it cannot be seen if the system is sustainable or not. In many cases of these simulation results the community kicks-off with good performance which soon gets dropped due to the unsuitable initial conditions. Fig. 1a and 1b are more informative regarding sustainability. Note that the oscillation which can be seen on Fig. 1b (and on other figures in later sections) are due to the fact that it shows average values of 5 independent runs and it is a clear indication of the fact that the actual configuration is unstable. One can observe performance decrease in case of 40% and 80% of rich peers. If the initial proportion of rich peers are set to be 80% the system crashes if the file size is larger than 10. We get a slightly better situation with 40% initial rich proportion. In this case the system is sustainable with $F = 20$, however, we have lower average performance than in those systems with higher proportion of rich peers. Having 40% rich peers and $F = 30$, the system does not perform in a stable manner, as it can be seen in the minimum throughput (equals to 195) or the average throughput which is as low as 705. The system crashed if $F > 30$.

All the other cases lead to sustainable community. As it was also shown in [19], $F = 10$ and 80% initial rich proportion leads to an unstable system in which the

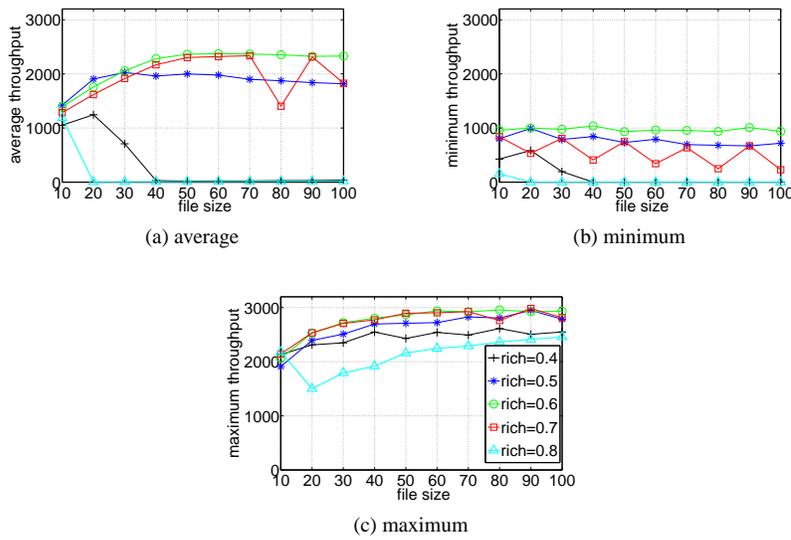


Fig. 1: Performance measures (homogeneous peers, default case)

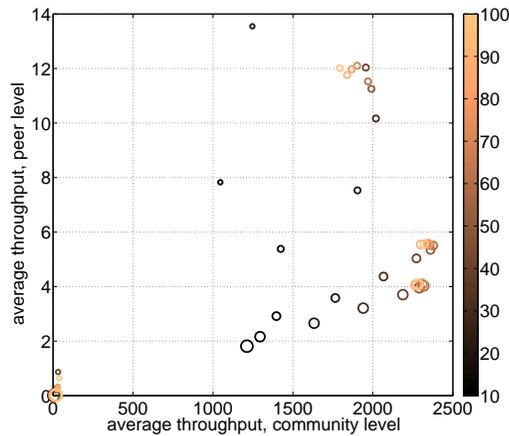


Fig. 2: Average throughput: community versus peer level (homogeneous peers, default case); bigger circles represent larger proportion of rich peers, colors represent file size

outcome can be both sustain and crash. Now, we can see that increase in the file size definitely ends up in crashed status.

Up until now, we have discussed the consequences of changing parameters regarding the whole system. It is also interesting to see what is the effect of these parameter changing on the individual user’s performance. Fig. 2 shows the follow-

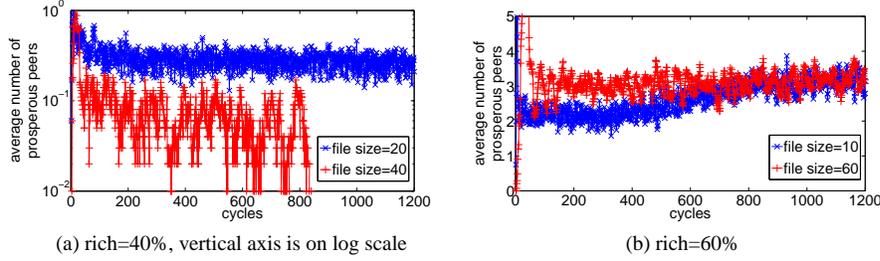


Fig. 3: Measurement of $1/S \cdot \sum_{\ell} |X_{\ell}| + |Y_{\ell}|$ in different scenarios, (homogeneous peers, default case)

ing. On the horizontal axis the average throughput of the whole system is shown, and the vertical axis shows the average throughput of a peer normalized with the file size. Bigger circle indicates larger proportion of rich peers. Colors of the circles represent the file size. What we can conclude from Fig. 2 is that individual performance and system-wide performance cannot be maximized together. As we already know from Fig. 1a the average throughput at community level is maximized in case of $F = 70$ and $R = 0.6$. Now we can also see that the best individual performance is guaranteed if both R and F are small.

The quantitative analysis of the set of prosperous peers can be seen on Fig. 3. Different scenarios were investigated. On Fig. 3a we have $R = 0.4$. As we already know, in this case we can have different outcome depending on the file size. The average number of prosperous leechers and seeders stays at a constant level around 0.2 if $F = 20$, which seems to be just enough to keep the system sustainable. On the other hand, at $F = 40$ we have less amount of thriving peers, which eventually lead to crunch. From these results we form the conjecture that the value of $1/S \cdot \sum_{\ell} |X_{\ell}| + |Y_{\ell}|$ must always be kept above 0.1 in order to assure sustainability. On the other hand, from the average number of prosperous peers one cannot predict the actual performance of the system. This is demonstrated on Fig. 3b on which we can notice that after about 600 cycles the two tested community is indistinguishable, however, they do have different average performance as we have learnt from Fig. 1a.

5.1.2 Homogeneous peers, constrained leecher upload

If we want to find explanation about crash or crunch in a BitTorrent community, it is natural to invest into the tracking of how the credit flows through the system. By definition, crunch is the situation when peers have no enough credit to start downloading and crunch is the opposite when peers have too much credit so they are not motivated to serve as uploaders. Thus, it does matter what the actual credit distribution among the peers is. Closer look into the different simulation runs reveals that sometimes peers can collect exceptional amount of credit while leeching and remaining rich after finishing the download, thus they can immediately start downloading the next file. This situation can be considered as “stealing” credit from the seeders in the

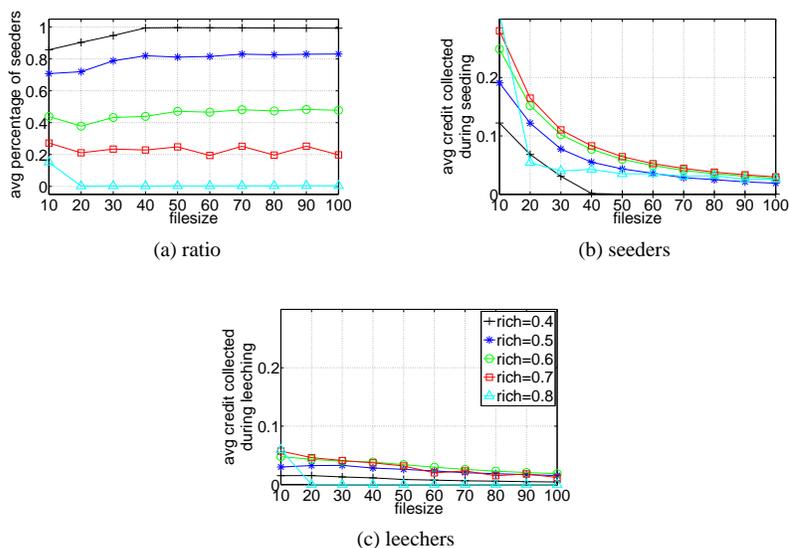


Fig. 4: Collecting credits (homogeneous peers, constrained leecher upload)

swarm. For this reason it is worth inspecting the proportion of seeders and leechers and the amount of credit these two types of peers can collect in each round. In order to see whether the leechers' credit absorption has any impact on the sustainability we performed simulations in which the credit collection of leechers are limited to the amount of the file size F .

Fig. 4a shows the average proportion of seeders under different scenarios given by different file size and by the initial amount of rich peers. On Fig. 4b we can see the average amount of credit a seeder can collect. Note that this amount is normalized with the actual file size of the system. Fig. 4c shows the same for a leecher. Comparing these results with those we obtained when the credit collection of leechers are not restricted we notice no significant differences.

Thus, we conclude that putting limit on the credit absorption does not have effect on the peers credit collection in different phases of the system. Regarding the performance measures, we do not get different results there either.

5.1.3 Heterogeneous peers, default case

Now we are interested to see the performance of the community under the conditions of different file size and heterogeneous peers bandwidth. In the following simulations peers were associated, uniformly at random, with either default download bandwidth (which is 8 pieces per round) or with slow download bandwidth equal to 2 pieces per round.

Comparing the results obtained we get quite similar figures than those from Section 5.1.1, thus they are omitted. In the following we describe the differences between the two scenarios. Regarding the performance measures (average, minimum

and maximum throughput) one notices differences only at 40% and 80% proportion of rich peers. If the system is started with 40% rich peers, the minimum and maximum throughput do not change significantly, while the average throughput increases for $F = 30$ and $F = 40$. This is actually the consequence of the fact that while with these parameters we get unstable and crashing systems in case of homogeneous bandwidth, introducing bandwidth heterogeneity leads to sustainable community at $R = 0,4$ and $F = 30$. With $F = 40$ we sometimes get simulation runs in which the transactions get ceased in later cycles. We have even more noticeable differences at 80% of rich peers, where the homogeneous systems crashed at $F = 20$ or $F = 30$. Heterogeneous bandwidth can play role in sustainability in these cases, however, the lower average performance clearly indicates that the final state can be sustain as well as crash.

Results of credit collection confirm the previous findings. With 80% rich peers and $F = 20$ or $F = 30$ we have increased values in the credit collection of both leechers and seeders compared to those shown on Fig. 4b and 4c, respectively.

The results we obtain with the heterogeneous bandwidth set up are promising as they indicate that more realistic scenarios lead to more stable systems. We can report that the average percentage of seeders are slightly higher under these circumstances compared to those shown on Fig. 4a.

Comparing the individual performance against community level performance, basically we obtain similar trend as for homogeneous peers (Section 5.1.1). The only noticeable difference is that individual performance can be higher in this case at low R and F values.

5.1.4 Heterogeneous peers, constrained leecher upload

Similar to Section 5.1.2, in these simulations we constrain the amount of pieces that leechers can upload to each other. As we have seen, this limitation had no effect on the performance in case of homogeneous system. However, restricting the upload amount in communities with heterogeneous peers leads to smaller average, minimum and maximum throughput values compared to the default case. On the other hand, we observed that these systems tend to behave like the ones we examined in Section 5.1.1 (i.e. homogeneous bandwidth without upload limit).

5.2 Effect of initial credit amount and distribution

In the following we investigate how the initial amount of credit plays any role in the sustainability of a BitTorrent community. Although we use the terminology 'initial amount', the different scenarios we study here resemble actual situations in real-life BitTorrent communities after so-called *freeleech* period [14]. During these periods peers do not need to spend credit for download. This is basically credit injection into the system, which can lead to situations considered in the followings.

In the different scenarios the initial credit amount is increased by 5% steps, starting from $N \cdot F$ multiplied by the proportion of rich peers, where $F = 10$. All in these runs, the initial credit amount of rich peers are usually larger than the amount needed

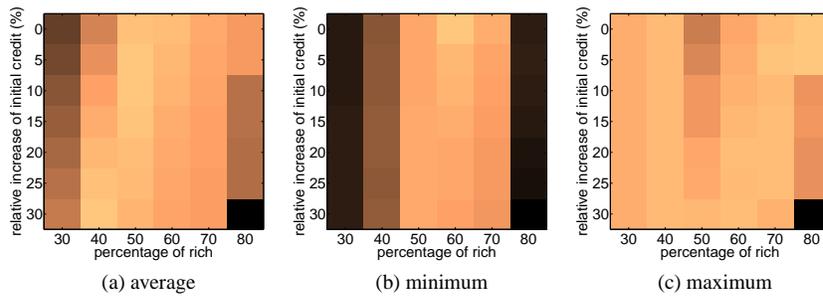


Fig. 5: Throughput (homogeneous peers, default case)

to download one file, and the poor peers can have credit as well. Note that in the previous sections poor peers always had zero credit at the beginning of the simulations.

5.2.1 Homogeneous peers, default case

Proceeding with similar arrangement of simulation runs as we have done in Section 5.1, let us start having a look on results obtained in systems of peers having homogeneous download capacity. The results are shown as heat maps, on which the horizontal axis represents the initial proportion of rich peers and the vertical axis indicates the increase of initial credit in the system. Thus the top row shows the same results we obtained in the previous section with $F = 10$. The heat maps of Fig. 5 show the mean, the minimum and the maximum throughput, averaged over 5 runs for each scenario. Values are associated with colors: darker rectangle indicates lower value, lighter color corresponds to higher value. Note that 10%, 20% and 90% initial proportion of rich peers always lead to unsustainable system, thus these results will not be discussed.

Considering the average throughput values on Fig. 5a one can notice an interesting diagonal shape of lighter colored (higher) values. This shows that the average performance can be increased with either (i) higher initial credit in case of lower proportion of rich peers, or (ii) lower initial credit in case of higher proportion of rich peers.

In case we consider the minimum throughput, see Fig. 5b, then we see again the diagonal pattern, though with lower values. It is worth noticing that while credit injection in case of 30-50% of rich peers can increase the average throughput, it does not have effect on the minimum throughput. The maximum of the minimal throughput is reached when the system starts at $R = 0.6$ together with no extra credit.

Results for the maximum throughput are shown on Fig. 5c. Remark that these must be considered with caution. For example, starting with $R = 0,8$ and 10% extra credit we get very high values, however, the system is very unstable, it can easily crash.

It can be clearly seen on Fig. 6 that the usual trade-off between individual and community performance is present in these systems. However, we can also notice

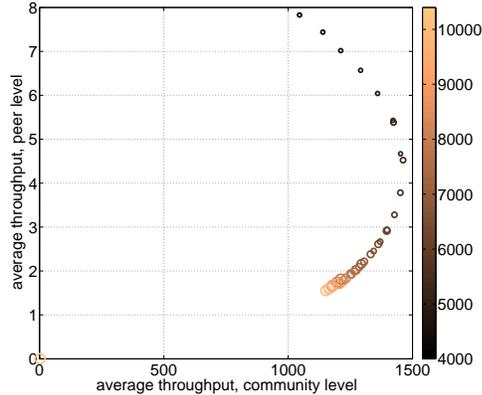


Fig. 6: Average throughput: community versus peer level (homogeneous peers, default case); bigger circles represent larger proportion of rich peers, colors represent initial credit amount

that there are configurations which result in systems with decent performance with respect to both measures. A very clear trend can be observed: higher amount of extra credit or higher R value definitely lowers the individual performance.

5.2.2 Homogeneous peers, constrained leecher upload

As before, we also performed simulation runs with putting limit on the credit collection of leechers so that during a download they cannot earn more credit than the size of the file. Results are shown on Fig. 7. It can be noticed that the systems with this feature are more sensitive, for example, the previously stable system with 70% rich peers crashes if it is started with too much extra credit. This is even more emphasized on Fig. 8, where the individual performance is compared with that of the whole community. We have much more circles at value 0 on both axes. Moreover, those circles corresponding to higher proportion of rich peers together with higher extra credit tend to have lower values.

Nevertheless, this setup has some interesting results concerning the minimum or maximum throughput values. Starting with 50-60% rich peers and some extra credit, we get higher performance compared to the standard case.

5.2.3 Heterogeneous peers, default case

In the previous experiments we have seen that systems with heterogeneous bandwidth can have higher performance. The main difference to the homogeneous case is that community with 80% initial rich peers besides having higher average and maximum throughput, it also has better stability. This performance increase includes that we do not have average throughput below 1,000 at the community level.

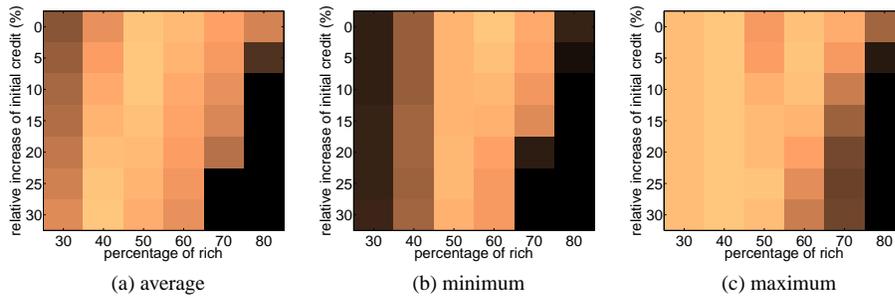


Fig. 7: Throughput (homogeneous peers, constrained leecher upload)

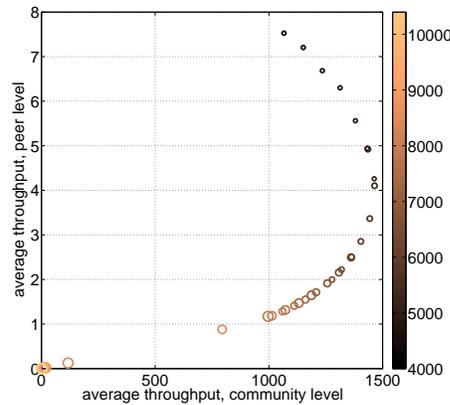


Fig. 8: Average throughput: community versus peer level (homogeneous peers, constrained leecher upload); bigger circles represent larger proportion of rich peers, colors represent initial credit amount

5.2.4 Heterogeneous peers, constrained leecher upload

Applying limit on the credit collection of leechers, we obtain similar results as with unlimited case. Although the average throughput slightly decreases, between 20% and 60% of rich peers usually leads to higher maximum throughput, and at $R = 0,5$ the minimum throughput is higher. Comparing the average performance at the community level against the individual level, we can report again that there is a cost for higher throughput at the community level: the individual performance drops with about 0,5 (the actual numbers are quite similar to those shown on Fig. 8) Moreover, we get lots of simulation results with 0 throughput value when the proportion of rich peers are too high, thus we have more sensitive systems in this case.

6 Summary

We have considered a parameter space of BitTorrent credit systems in order to study the conditions for sustainability. Several kinds of systems were taken into account including peers with homogeneous and heterogeneous peers bandwidths, and limited credit collection during leeching phase. We performed two case studies: effect of file size and impact of initial credit amount distribution.

Our results show that the sustainability does not only depend on the initial proportion of the rich peers (leechers), but also on the file size. We have demonstrated that larger file size has three important effects. First of all, it decreases the range of sustainable configurations to 50%–70% proportion of rich peers; in the sustainable range it slightly increases the average throughput measured at the community level; however, it can decrease the performance of individual peers.

We have also shown that increasing the initial amount of credit in the system can lead to higher average throughput in the community, but only up to a certain level. Using different parameters can increase the performance of individuals, but this must be done with care, adapting to the needs in the system. The minimum and maximum throughput of the system can be increased with limit on the leechers credit collection.

Our results contribute to the deeper understanding of the side-effects of credit-based incentive mechanisms of private BitTorrent communities. Given such a community, it is much more clear now when and what kind of intervention method should be applied in order to assure sustainability and performance increase.

Acknowledgements The authors would like to thank the anonymous reviewers who have helped to improve the paper. This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013). T. Vinkó was supported by the Bolyai Scholarship of the Hungarian Academy of Sciences.

References

1. Andrade, N., Santos-Neto, E., Brasileiro, F., Ripeanu, M.: Resource demand and supply in BitTorrent content-sharing communities. *Computer Networks* **53**, 515–527 (2008)
2. Axelrod, R.M.: *The evolution of cooperation*. Basic books (2006)
3. Chen, X., Chu, X., Li, Z.: Improving sustainability of private p2p communities. In: *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN'11)*, pp. 1–6 (2011)
4. Chen, X., Jiang, Y., Chu, X.: Measurements, analysis and modeling of private trackers. In: *IEEE Tenth International Conference on Peer-to-Peer Computing (P2P'10)*, pp. 1–10 (2010)
5. Chu, X., Chen, X., Jia, A., Pouwelse, J., Epema, D.: Dissecting Darknets: Measurement and Performance Analysis. *ACM Transactions on Internet Technology*, **13**, 1–25 (2014)
6. Cohen, B.: *Incentives Build Robustness in BitTorrent*. In: *Workshop on Economics of Peer-to-Peer Systems*. Berkeley, USA (2003)
7. Garbacki, P., Epema, D., van Steen, M.: An amortized tit-for-tat protocol for exchanging bandwidth instead of content in p2p networks. In: *First International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07)*, pp. 119–128 (2007)
8. Hales, D., Rahman, R., Zhang, B., Meulpolder, M., Pouwelse, J.: Bittorrent or bitcrunch: Evidence of a credit squeeze in bittorrent? In: *18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE '09)*, pp. 99–104 (2009)
9. Jia, A., Rahman, R., Vinko, T., Pouwelse, J., Epema, D.: Fast download but eternal seeding: The reward and punishment of sharing ratio enforcement. In: *IEEE International Conference on Peer-to-Peer Computing (P2P'11)*, pp. 280–289 (2011)

10. Jia, A., Rahman, R., Vinkó, T., Pouwelse, J., Epema, D.: Systemic risk and user-level performance in private p2p communities. *IEEE Transactions on Parallel and Distributed Systems* **24**(12), 2503–2512 (2013)
11. Jia, A., Chen, X., Chu, X., Pouwelse, J., Epema, D.: How to Survive and Thrive in a Private BitTorrent Community. *Lecture Notes in Computer Science* **7730**, 270–284 (2013)
12. Jia, A., Chen, X., Chu, X., Pouwelse, J., Epema, D.: User behaviors in private BitTorrent communities. *Computer Networks* **60**, 34–45 (2014)
13. Kash, I.A., Friedman, E.J., Halpern, J.Y.: Optimizing scrip systems: Efficiency, crashes, hoarders, and altruists. In: *Proceedings of the 8th ACM Conference on Electronic Commerce, EC '07*, pp. 305–315. ACM, New York, NY, USA (2007)
14. Kash, I.A., Lai, J.K., Zhang, H., Zohar, A.: Economics of BitTorrent communities. In: *Proceedings of the 21st International Conference on World Wide Web (WWW'12)*, pp. 221–230. ACM, New York, NY, USA (2012)
15. Kaufman, G.G.: Banking and currency crisis and systemic risk: A taxonomy and review. *Financial Markets, Institutions & Instruments* **9**(2), 69–131 (2000)
16. Li, Q., Qin, T., Guan, X., Zheng, Q.: Analysis of users behavior and resource characteristics for private trackers. *Peer-to-Peer Networking and Applications* pp. 1–15 (2014)
17. Meulpolder, M., D'Acunto, L., Capotua, M., Wojciechowski, M., Pouwelse, J., Epema, D., Sips, H.: Public and private BitTorrent communities: A measurement study. In: *Proceedings of the 9th international conference on Peer-to-peer systems (USENIX Association, 2010)*, pp. 1–10 (2010)
18. Qiu, D., Srikant, R.: Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. In: *Proceedings of ACM SIGCOMM* (2004)
19. Rahman, R., Hales, D., Vinkó, T., Pouwelse, J., Sips, H.: No more crash or crunch: Sustainable credit dynamics in a p2p community. In: *International Conference on High Performance Computing and Simulation (HPCS'10)*, pp. 332–340 (2010)
20. Ramachandran, A., Feamster, N.: Bitstore: An incentive-compatible solution for blocked downloads in bittorrent. In: *2nd Joint Workshop on Economics of Networked Systems and Incentive-Based Computing* (2007)
21. Turner, D.A., Ross, K.W.: A Lightweight Currency Paradigm for the P2P Resource Market. In: *7th International Conference on Electronic Commerce Research* (2004)
22. Zghaibeh, M., Anagnostakis, K.G., Harmantzis, F.C.: The behavior of free riders in bittorrent networks. In: *Handbook of Peer-to-Peer Networking*, pp. 1207–1230. Springer (2010)
23. Zhang, C., Dhungel, P., Wu, D., Liu, Z., Ross, K.W.: Bittorrent darknets. In: *Proceedings of the 29th Conference on Information Communications, INFOCOM'10*, pp. 1460–1468. IEEE Press, Piscataway, NJ, USA (2010)