The Seeder Promotion Problem: Measurements, Analysis and Solution Space

Sebastian Kaune*, Gareth Tyson†, Konstantin Pussep*, Andreas Mauthe† and Ralf Steinmetz*
*Technische Universität Darmstadt, †Lancaster University

***Abstract*—BitTorrent has become the de-facto standard for peer-to-peer content delivery, however, it has been found that it suffers from one fundamental problem: *the long-term availability of content*. Previous work has attributed this to what is termed the *seeder promotion problem* in which peers refuse to continue serving content after their own download has completed. As of yet, no deployed solution exists to this problem.**

In this paper, we objectively investigate the solution space for dealing with the seeder promotion problem. Specifically, both single-torrent and cross-torrent approaches are investigated to ascertain which is superior based on three key metrics: availability, performance, and fairness. To achieve this, two large-scale BitTorrent measurement studies have been performed which include 46K torrents and 29M users. Through these, we first quantify the seriousness of the seeder promotion problem before exploiting the data logs to execute accurate trace-based simulations for the different solutions considered. Using the results, we ascertain and describe the different trade-offs between the four general solutions: extending seeding times, cross-torrent bartering, local persistent histories, and global shared histories.

We find that single-torrent solutions are profoundly impractical when considering the user behaviour observed in our studies. In contrast, we discover that the different cross-torrent approaches can offer a far more effective solution for satisfying (to varying degrees) the need for high availability, good performance, and fairness between users.

I. Introduction

BitTorrent [1] has become a de-facto standard for scalable content distribution over the Internet. However, despite this success, BitTorrent suffers from one major problem: *long term file availability* [2]. It can be observed that after a relatively short period of time many torrents cease to offer a fully available file. This occurs because vital file pieces become missing therefore preventing a peer from reconstructing an entire file.

The cause of this issue has been termed the *seeder promotion problem* [3]. Peers within a swarm that have completed their download are termed *seeders* while those that are in the process of downloading are termed *leechers*. The seeder promotion problem therefore occurs when leechers refuse to become seeders resulting in the likelihood that certain file pieces cease to be collectively available in the swarm. This means that torrents without seeders usually offer unavailable files that have stalled performance.

In general, two categories of solutions exist for addressing this problem. The first group is termed *single-torrent* solutions, which involve peers within a *single* swarm cooperating to ensure the availability of rare pieces. As such, incentives (such as file bundling [4]) are used to encourage users to remain seeding for longer periods of time. In contrast, the second group is termed *cross-torrent* solutions and involves the peers of *multiple* swarms cooperating to ensure the availability of rare chunks. This is motivated by the observation that more than 85% of the users in BitTorrent systems participate in multiple torrents [2]. Cross-torrent collaboration therefore exploits this observation to build collaboration between swarms so that users can offer files they have previously downloaded to other torrents while also downloading files of their own interest. As such, cross-torrent solutions aim to build incentives that allow users to make contributions and receive rewards agnostic to the peers or swarms they are performed in.

A small number of key approaches exist for building cross-torrent incentives [5]–[7], however, so far, they have not been properly evaluated and compared. This paper aims to objectively investigate the feasibility of cross-torrent approaches while contrasting them against single-torrent alternatives. To achieve this, two detailed measurement studies of BitTorrent have been performed to (i) understand the seeder promotion problem and (ii) gain sufficient data to allow an accurate trace-based evaluation of the different approaches. Therefore, with this trace-based information, three key cross-torrent solutions are inspected through a detailed simulation study. The intention of this study is to answer the following research questions,

- Is it possible to solve the seeder promotion problem through a single-torrent approach?
- Can cross-torrent solutions offer superior (i) availability and/or (ii) performance to single-torrent alternatives?
- What are the trade-offs between the different single-torrent and cross-torrent solutions?

The rest of this paper is structured as follows; first, the methodology of the measurement studies is detailed. Following this, in Section III we utilise the measurement data to explore the scale and validity of the seeder promotion problem in regards to BitTorrent's availability. In Section IV, we then explore the cross-torrent solution space to highlight three key approaches for building such incentives. In Section V, these different approaches are evaluated based on three primary metrics: availability, performance and fairness. We present the related work in Section VI and conclude in Section VII.

II. Measurement Methodology

To study the seeder promotion problem, it is important to understand the behaviour of peers in real-world torrents. To achieve this we have conducted large-scale measurement

978-1-4244-7116-4/10/$26.00 ©2010 IEEE

studies of torrents indexed by the Mininova website[1] [8]: two microscopic studies and a single macroscopic one.

***Microscopic Crawling*:** To truly understand unavailability in BitTorrent it is necessary to be able to view the distribution of pieces within any given swarm. Without this, it is only possible to infer availability based on alternative related metrics such as the number of seeders. To gain this information we therefore developed and deployed a BitTorrent crawler that can investigate swarms on a microscopic level. The crawler operated from July 18, 2009 to July 29, 2009 (*micros-1*) and then again from August 19, 2009 to September 5, 2009 (*micros-2*). In a time resolution of 10 minutes, this crawler requested (using PEX [9]) the piece bitmaps and routing tables entries from each peer in the swarm. To prevent the blacklisting of monitor IP addresses, the request process was distributed over 20 nodes hosted on the Emulab [10] testbed. All information was logged creating 7 GB and 12 GB of data in micros-1 and micros-2, respectively. For the *micros-1* study, the crawler followed 255 torrents appearing on Mininova after the first measurement hour. In these torrents, we observed 246,750 users. The *micros-2* dataset contains information from 577 torrents and 531,089 users.

***Macroscopic Crawling*:** The macroscopic measurements provide detailed insight into the distribution of data pieces within the swarm, as well as between different peers. However, due to scalability issues it is impossible to perform such detailed measurements on a extremely large-scale (e.g. $>$ 1000 torrents). To complement these results we therefore also implemented a crawler that followed every torrents published on the Mininova website in a much higher-level manner after December 09, 2008 for a period of 38 days. This crawler repeatedly requested, from multiple sites in Europe, information regarding each torrent's number of seeders and leechers. By doing so, we were able to continuously discover 98% of the online peers reported by the tracker. With this information, it is possible to use a similar approach to [6], [11] to identify different users; we assume a user comes online the first time it is reported by the tracker, and stays online until it is no longer included in the tracker report. This study allowed us to gain an extremely large number of measurements regarding such things as peer arrival patterns, seeder:leecher ratios and torrent sizes. This information can subsequently be correlated with our smaller-scale microscopic measurements to derive such things as the scale of seedless states and the causes for seedless states occurring. Our final macroscopic dataset consisted of reports from 46,227 torrents and 29,066,139 users.

In the remainder of this paper, we will use both kinds of measurements simultaneously for our analysis.

III. The Impact of Seeders in BitTorrent

Before investigating the solution space of the seeder promotion problem it is important to understand its real-world characteristics. To this end, we first inspect the measurement results to ascertain (i) the nature of the seeder promotion problem, (ii) the effect of seeder departures and, finally, (iii) the scale of the problem.

[1]It is the most popular repository of torrents based on the Alexa Ranking.

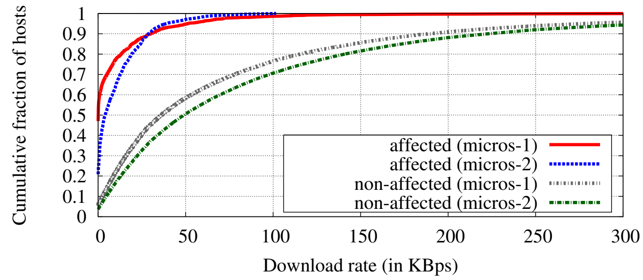

Fig. 1. Comparison of download rates of peers that are affected by the lack of seeders and those that are not.

A. The Seeder Promotion Problem

The effectiveness of BitTorrent can largely be attributed to its rate-based tit-for-tat incentive mechanism that encourages users to contribute resources to achieve higher performance. Despite this, however, it can be observed that many torrents do not seem to benefit from this. Instead, a massive proportion of torrents ($\approx$ 40%) achieve extremely low performance with few users being able to download the file successfully.

The reason for this significant divergence in performance is the so called *seeder promotion problem*. This occurs because users are given no incentives to remain online to serve a file after their download has completed (i.e. to act as seeders). It has been observed that seeders play a vital role in BitTorrent's performance as they (i) provide resources without consuming any [12], and (ii) ensure that a complete copy of the file remains in the swarm [6]. We believe that it is the latter point that is most vital for BitTorrent's performance as a swarm without a full copy of the file is implicitly unavailable, even if it is only one chunk that is missing. As such, we consider BitTorrent's performance to be closely linked to its availability.

B. Effects of Seed Departure

The first step we take is to investigate the effect that seeder departures have on performance. Fig. 1 depicts the cumulative distributions of the download rates of users operating in (i) torrents with highly intermittent seeders and (ii) torrents with one or more seeders. A torrent with intermittent seeders is one that has, at some point during our measurement study, suffered from a seedless state. This is most likely an older, less popular torrent.

It can be observed that performance heavily degrades for users that are affected by a lack of seeders: the median download rate is less than 3 KBps. As a consequence, we observed extremely high download abortion rates in both microscopic measurement studies ($\approx$ 89%). Interestingly, we also observe a chain reaction; as soon as the last seeder leaves the torrent, download rates drop and many users choose to abort. Consequently, these users never become seeders and the cycle is prolonged until a past seeder returns.

To validate that this performance degradation is caused by unavailability, we also inspect the seedless torrents. Fig. 2

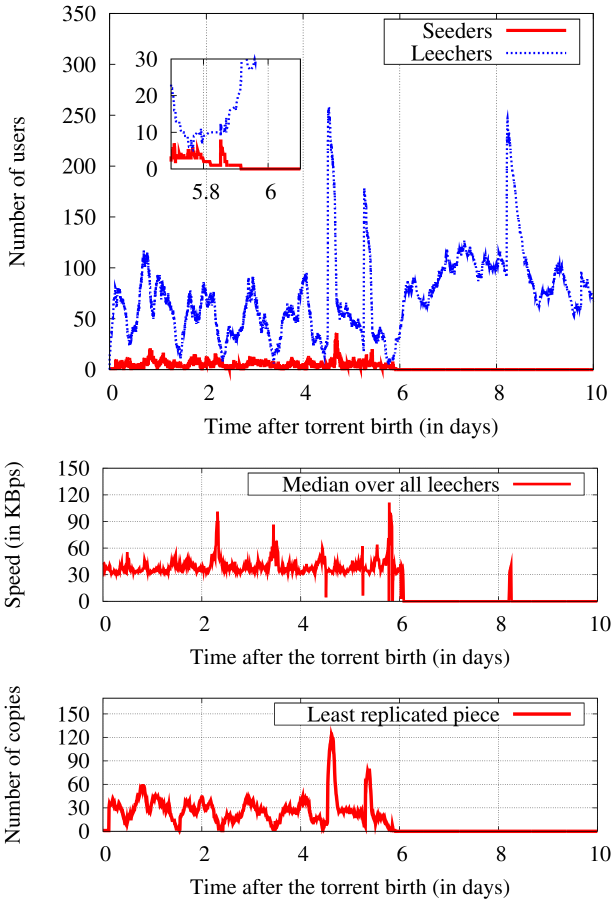

Fig. 2. Snapshot from a torrent in our microscopic trace.

shows the correlation between the number of seeders, average download rate and number of the least replicated piece for an example torrent[2]. It can be seen that as the last seeder departs on the fifth day, both the average download rate and least replicated piece drops to zero. This subsequently means this torrent is unavailable. This behaviour is observed in all affected torrents; from this it can be inferred that the three issues are dependent. As such, it is evident that any solution must consider these three aspects as equally important.

C. Scale of Seedless States

It is evident from the previous results that a lack of seeders results overwhelmingly in poor performance. The next step is to try to understand the behaviour of seeders so that the solution space can be explored. The measurements reveal that the existence of seedless torrents can be attributed to the exponentially increasingly inter-arrival times of users. In essence, this means that the seeders do not stay online for long enough to assist in the production of new seeders. We find that over 75% of seeders remain online for less than 4 hours, while inter-arrival times between users quickly exceeds 10 hours for more than 45% of torrents. This results in more than 38% of torrents losing their seeders within the first month, out of which 72% are without a seeder after only 5 days. Similarly, we find that more than 45% of the torrents suffer from a lack of seeders for half of their monitoring time. To exemplify the scale of this, in 50% of the torrents observed for periods longer than 30 days, no seeder was available for more than 16 days.

Clearly, the problem is wide-spread with more than 9.68 million users participated in torrents with intermittent seeders. Out of these users, more than 1.59 million were directly affected by the seedless states.

[2]Similar behaviour has been observed in all seedless torrents.

IV. Cross Torrent Solution Space

The previous section has explored and validated the presence of the seeder promotion problem with regards to availability and performance. The traditional BitTorrent paradigm operates on an individual torrent basis, however, this section now inspects the cross-torrent solution space which builds incentives across multiple torrents.

A. Overview and Motivation

The results of previous section reveal that the performance of BitTorrent is largely dictated by the presence of seeders. Without seeders, a torrent typically does not possess all pieces resulting in download stalling.

The most intuitive solution to this is to take a single-torrent approach that encourages seeders to remain in a swarm, perhaps using out-of-bands mechanisms (e.g. monetary reward). However, an alternative is to find ways of encouraging cooperation between multiple torrents. A cross-torrent solution involves incentivising users to cooperate with the *system* as opposed to individual *torrents*. This approach is motivated by observations from our macroscopic trace that shows 51% of the users join multiple torrents (4.98 on average). We have further found that seeders frequently rejoin swarms after they have left, therefore providing conclusive evidence that the same peers rejoin the BitTorrent system multiple times while still possessing their previously downloaded files.

To highlight the principles of a cross-torrent solution, imagine a user who joins torrent X at some point in time and completes the download as shown in Fig. 3. This user may very well join another torrent Y at a later point in time. When this occurs, the node could also theoretically persist as a *replica* for torrent X as shown by the dashed bold sections in Fig. 3. As such, the seeder promotion problem would be addressed by utilising replicas as opposed to traditional seeders (although in practise these are very similar). To inspect the feasibility of this, Fig. 4 shows the number of online nodes in the 'Movies' category alongside the number of potential online nodes that could act as replicas. Evidently, there is a large pool of untapped resources that could be exploited; in fact, we find that for all torrents there is at least one available replica after 36 hours.

So far it has been shown that there is real-world potential for utilising cross-torrent solutions for addressing the seeder promotion problem. However, as of yet, there exists no deployed solution for incentivising users to cooperate in such a process. In fact, the need to divert upload resources from a

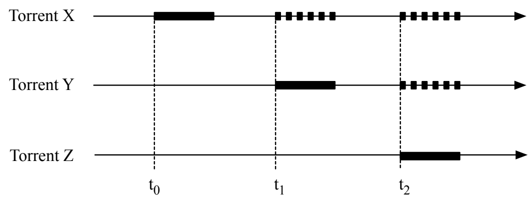

Fig. 3. Exemplary lifetime of a peer. The long arrows represent torrents. The bold sections represent intervals when the exemplary peer is active in the corresponding torrents as a leecher. The dashed bold sections represent intervals when the peer could resume seeding in the corresponding torrents in case there were any incentives for it to do so.

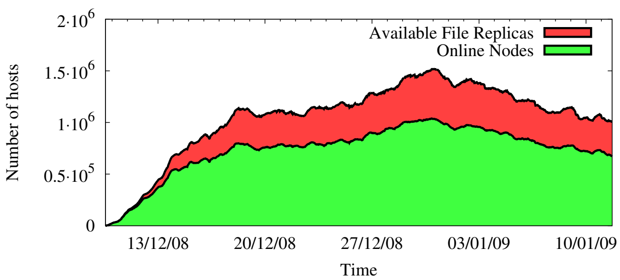

Fig. 4. Online users and available file replicas in the category 'movies' in our macroscopic trace.

node's current torrent would disincentivise cross-torrent collaboration because it would decrease the probability of a node being unchoked for its own content download. It is therefore important to build robust incentives alongside any cross-torrent protocols; to this end, this section now outlines three abstract cross-torrent incentive approaches that can encourage users to act as replicas.

B. Rate-Based Bartering

The most straight-forward approach is to extend BitTorrent's tit-for-tat mechanism to operate across multiple torrents. This involves peers bartering with each other for content regardless of what swarm they operate in. This could work as follows: assume that user A has previously downloaded torrent X (fully or partially) and is a leecher in torrent Y. User B, on the other hand, has obtained torrent Y earlier and is now a leecher in torrent X. Both A and B could mutually exchange chunks while still conforming to BitTorrent's tit-for-tat strategy.

This approach has the advantage that it is instant, based on personal experiences and does not induce any overhead to exchange information about cooperation across torrents. However, it also has the limitation of needing to locate other peers with shared interests subsequently restricting the applicability of the approach in any circumstances where such reciprocation cannot be found.

C. eMule-like Volume-Based Persistent History

Traditional tit-for-tat and cross-torrent bartering are based on rate-based incentives that are implemented in real-time (i.e. contributions and rewards are instant). An alternative is to base incentives on long-term persistent observations based on total data volume, as exemplified by eMule [13]. In eMule, peers *locally* maintain a persistent history of the contributions made by each user, agnostic to which file and to the time the contribution is made. Subsequently, peers would show preference to piece requests from users with higher contribution ratios. As such, peers are encouraged to act as sources for as many files as possible so that they can build up a positive reputation.

This approach has the advantage that it might increase the probability of locating shared interests (as with bartering) because incentives become long-term and persistent rather than instant. Importantly, this can also be achieved without introducing any communications overhead or the threat of using third parties in the process. Unlike bartering, the process is also detached from time, thereby allowing peers to claim back contribution at a later date. However, persistent histories still require repeated interactions between peers, possibly resulting in restricted applicability.

D. Indirect Reciprocation

The previous two approaches rely on direct observations that are stored locally. An alternative is to use persistent history information that is agnostic to individual peers. As such, a peer would be able to make a contribution to peer X and receive the reward from peer Y seamlessly. This would require some form of reputation infrastructure that can reliably store information about a given peer's 'balance'; a prominent example of this is a digital monetary system [14]. Alternatively, there are many approaches that allow reputations to be propagated amongst nodes by using different levels of indirection, e.g., [7], [15]–[17].

Indirect reciprocation has the advantage of detaching incentives from time, torrents and individual peers, thereby offering the 'purest' form of cross-torrent collaboration. This subsequently addresses the need for direct reciprocation (bartering) or repeated interactions (persistent histories). However, such an approach also introduces far greater complexity and overhead into the system, potentially negating these benefits in certain environments.

V. Evaluation

To evaluate the possible solutions, we use the BitTorrent simulator of Bharambe et al. [12]. Within this section we first detail the evaluation goals followed by an overview of the workloads used for the simulator.

A. Overview of Evaluation

1) Evaluative Aims: We do not aim to perform an implementational comparison between vanilla BitTorrent and the proposed approaches, e.g., regarding protocol overhead and technical aspects to realize either approach. This is out of the scope of this paper. The goal of our evaluation is *(i)* to find out whether the seeder promotion problem can be solved through a single-torrent approach and *(ii)* to shed light on the potential of the different cross-torrent alternatives. In particular, we wish to discover:

- Does cross-torrent collaboration increase file *availability* in torrents with ordinarily unavailable seeders?

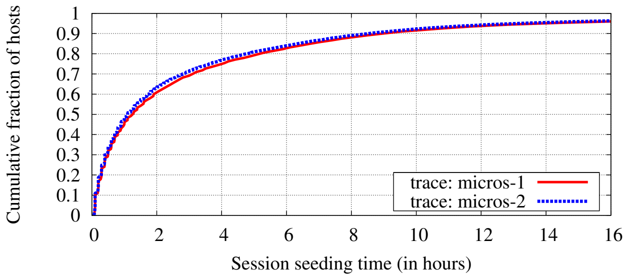

Fig. 5. Distribution of seeding times of BitTorrent users as obtained from our microscopic traces.

- What are the implications of this in regard to *download performance*?
- How effective are the sharing incentives in terms *fairness*? Does a peer that contributes more finish its download sooner?

To address these issues, we perform an extensive trace-driven simulation study using the data acquired by our crawlers. To benchmark the three cross-torrent approaches, we use as a baseline vanilla BitTorrent as deployed in real swarms today. In addition, we assume for our analysis a single-torrent approach capable of enlarging the measured seeding times of users by a factor of 2, 5, and 10.

2) Input to the Experiments:

Selecting the torrents: Our trace data encompasses tens of thousands of torrents over a period of several weeks, far more then the simulator is able to handle. Hence, we chose a random subset of hundred torrents from the set of torrents affected by seedless states with file sizes varying between 3-1500 MB and a per-torrent monitoring period of at least four weeks [3]. The logs of these torrents contains data of more than 45,000 downloads.

User behaviour: To model the access pattern of torrents, we do not use any artificial peer arrival function. Instead, we bring up peers according to the trace logs. To model the number of swarms that a peer joins we calculate the probability distribution over our entire data set. Any user that cannot download the file within 36 hours aborts the download[4]. The session time of a user consists of busy and idle periods. In *busy* period, the user is actively downloading until it reaches seeder state. Subsequently, the *idle* period begins in which the user remains until it quits the client. During this idle time, it serves already obtained content. To realistically model idle periods of BitTorrent users, we use the measurement data from our microscopic crawlings (cf. Fig. 5).

Speed distributions: To have a representative bandwidth distribution, we first associate each IP address with a country, using a freely available geolocation database [18]. Based on the country of origin, the Ookla database [19] provides us with the median down/uplink capacity of each user.

Failures in shared contribution histories: To represent information inconsistencies in the distribution of contribution histories (e.g. due to churn), when encountering a new user in the indirect reciprocity approach, the contribution history is only known with a probability of 0.9.

B. Availability

A file is considered unavailable if at least one of its pieces is not accessible within a swarm. This situation often coincides with a lack of seeders as seen in Sect. III-B. It means that any users attempting to download the file will fail; a prominent metric for measuring this is the abortion rate as most users are only prepared to wait a limited length of time during an unavailability period.

Fig. 6 shows the fraction of users that abort their downloads when utilising the different approaches. In addition, Tab. I gives an overview of the idle times of users and their seeding times on average. Note that in the cross-torrent variants, the idle time obviously differs from the seeding time. This is because cross-torrent collaboration allows users to seed in two or more torrents while being in busy and/or idle state.

The simulations show that $\approx 20\%$ of downloads were not successful in vanilla BitTorrent. This confirms our observation that nodes do not remain as seeders for long enough to overcome the exponentially increasing inter-arrival times of users. Worse, due to extremely long inter-arrival times (often exceeding 10 hours), a single-torrent approach even capable of increasing seeding times by a factor of 2 or 5 is limited in its success. To maintain persistent file availability (i.e. success rate $> 99\%$), the users must therefore stay on average 10 times longer after downloading. As such, to achieve availability, vanilla BitTorrent would require average seeding times of more than 34 hours.

The first cross-torrent approach inspected is *rate-based bartering*; the results show that this also fails to significantly improve availability. In fact, there is only a 2.78% improvement over vanilla BitTorrent. This occurs because cross-torrent bartering assumes that large numbers of peers operate in swarms with synchronous interests. The trace-based simulations show that this is, in fact, not an accurate assumption. The measurement study results also corroborate this finding; these show that the probability of bartering working in the real-world is below 0.1%. Therefore, the circumstances in which users can act as file replicas are very seldom due.

In contrast to these results, the other two cross-torrent approaches (the eMule-like and indirect reciprocity) are able to effectively maintain persistent file availability. As opposed to rate-based bartering, these solutions do not require immediate reciprocation. Instead, peers can claim back their rewards in the future and are thereby encouraged to act as a file replica in the hope of later gaining an advantage. In the case of eMule-like incentives this involves repeat interactions while in the case of indirect reciprocation this involves interacting with any peer. This approach of detaching incentives from time therefore perfectly addresses the availability issue.

[3] We have also experimented with higher/smaller amount of torrents. For conciseness, we report only the results from this representative sample. Our insights and conclusions are, however, consistent with other samples.

[4] We find through simulations that 36 hours is enough time to get a download success ratio over 99% in the presence of seeders for all access links and file sizes used in our experiments.

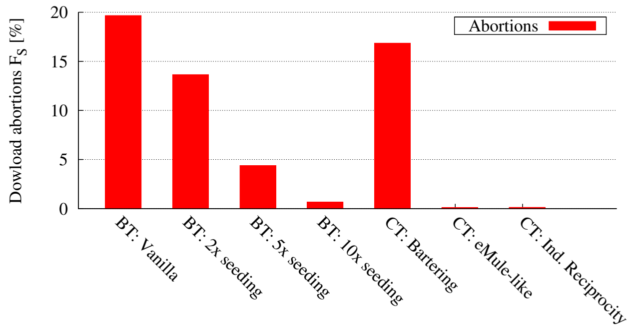

Fig. 6. Fraction of download abortions of the different approaches.

Variant	User stats		Metric	
	I (hours)	μ (hours)	D (KBps)	F (%)
BT: Vanilla	3.37	3.37	96.97	19.67
BT: 2x seeding	6.88	6.88	124.28	13.65
BT: 5x seeding	17.20	17.20	166.95	4.39
BT: 10x seeding	34.40	34.40	183.12	0.66
CT: Bartering	3.37	4.84	85.95	16.86
CT: eMule-like	3.37	10.36	51.45	0.11
CT: Ind. Reciprocity	3.37	9.51	130.81	0.13

TABLE I

OVERVIEW ABOUT RESULTS. I AND μ IS THE AVERAGE USER IDLE TIME AND SEEDING TIME, RESPECTIVELY. THE AVERAGE DOWNLOAD RATE ON A SYSTEM LEVEL AS WELL AS THE FRACTION OF DOWNLOAD ABORTIONS IS ABBREVIATED BY D AND F.

C. Performance

While some solutions have been shown to enable persistent availability, we have also found that users are highly sensitive to their perceived instant quality of service (c.f. Sect. III-B). Therefore, any solution must also maintain an acceptable download rate while improving availability. To study this, the instant average download rates in each torrent have been recorded when utilising the various cross-torrent solutions. Fig. 7 shows the cumulative distribution of these download rates.

It can first be observed that roughly 20% of the downloads in vanilla BitTorrent are below 1 KBps. As such, it can be considered that performance is unacceptably low. The reason for this is the poor availability observed in the 20% of torrents as discussed in the previous section. Clearly, both availability and performance in BitTorrent are inexorably linked: torrents that are unavailable also have low performance. Of course, as previously shown, this problem can be addressed by extending seeding times, thereby ensuring availability. Interestingly, this would also have the added benefit of increasing swarm resources as exemplified by the highest download rate in Tab. I (183 KBps for 10x seeding).

Considering the previous results, it is unsurprising that rate-based bartering also does not offer significant performance benefits. This is because it is essentially the same as vanilla BitTorrent (tit-for-tat) but with the ability to operate across different torrents. It has previously been shown that this does not really improve availability and consequently this results in poor performance due to the dependency of performance on availability.

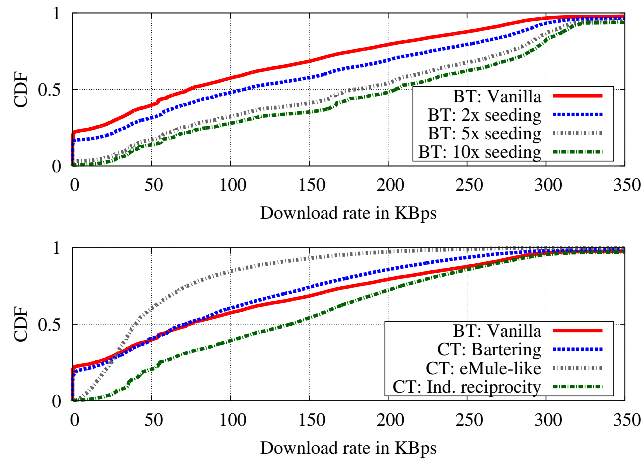

Fig. 7. Cumulative distribution functions (CDF) of the measured download rates. Top: Variants with lengthened seeding times. Bottom: Cross-torrent approaches.

The previous section found that eMule-like incentives do improve availability and therefore it is logical to assume that performance is also improved. In fact, this approach does result in a significant increase in nodes that find content available (>19%). However, the simulations show that this does not translate into performance improvements. Instead, users can access the files with high availability but with poor performance. This is consistent with most people's daily experience of using the eMule application. The reason for this is that eMule relies on repeat interactions by maintaining persistent records. A peer that contributes resources to another peer can therefore only recoup them if there is a later repeated interaction. This therefore creates incentives for sharing but prevents a peer from claiming back contributions from an arbitrary peer on many occasions.

The last cross-torrent solution inspected is indirect reciprocation which has already been shown to vastly improve availability. The results show that, unlike eMule-like incentives, this actually does translate into superior performance. In fact, data inspection reveals that 54% of the users would gain a performance boost of a factor of more than 4 when switching from the eMule-based approach to indirect reciprocation. The reason is that indirect reciprocation allows users to make contributions and claims them back from any user and any torrent without the need for repeated interaction. This means that a peer will receive superior performance from any peer if it, in return, offers resources to the system as a whole.

D. Fairness

Whereas the previous two evaluative metrics have looked at aspects that are vital for the continued success of BitTorrent, a further property that would also be desirable is *fairness*. This is defined by the amount of reciprocated data generated by contributions. For incentives such as tit-for-tat and bartering, reciprocation is immediate and can therefore be directly measured. For persistent contribution histories,

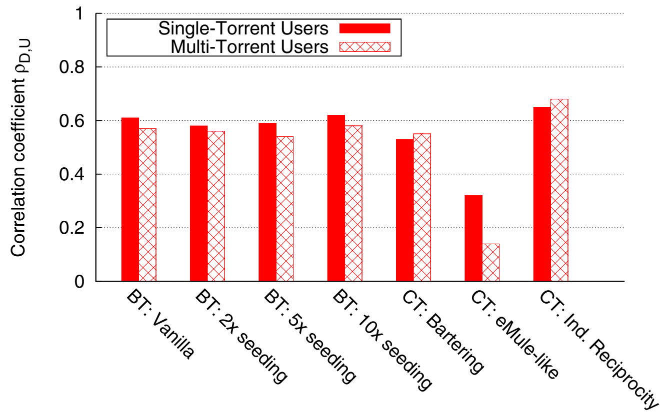

Fig. 8. Correlation between download and upload speeds for single-torrent and multi-torrent users.

however, peers may experience lengthy delays before receiving reciprocation. Simply considering a snapshot of the users' share ratio can therefore be a misleading measure because in the future this measure may change. For that reason, we consider the following fairness criteria: any peer x with an upload rate Up_x should get a higher download rate than any other peer y with an upload rate $Up_y < Up_x$.

To quantify the relation between the average upload rate of users (U) and their experienced download rate (D), we compute the correlation coefficient $\rho_{D,U}$ over all users that join (i) just a single torrent and (ii) multiple torrents, as shown in Fig. 8. In particular, differentiating users in these two groups allows us to quantify whether users persisting as file replicas benefit from their behaviour.

The single-torrent variants (vanilla and extended seeding) all show a positive correlation between (U) and (D). In fact, their correlations are very similar suggesting that they all offer a similar level of fairness. Clearly, reciprocation in BitTorrent is based on immediate rate-based observations and, as such, it is not surprising that the peer selection strategy does an effective job of matching users with similar capabilities [20].

Similarly, cross-torrent bartering offers a high level of fairness that is largely identical to vanilla BitTorrent. This is intuitive as it operates using BitTorrent's peer selection strategy with the added capability of being able to interact with peers in different torrents.

In contrast to these results, the eMule-like approach exhibits at best a weak correlation for single torrent users and no correlation at all for multi-torrent users. This suggests that users see poor returns when acting as file replica. The reason for this is twofold. First, the need for repeated interactions means that sometimes a peer will make a contribution without ever receiving any benefit in the future. This can occur due to permanent peer departures or, alternatively, due to bad luck on the part of the contributor. In fact, within the measurement study, only 19% of users ever meet each other repeatedly resulting in 81% of contributions being unclaimed.

Last, the indirect reciprocation approach offers a similar level of fairness to vanilla BitTorrent. However, when using indirect reciprocation a peer makes unchoking decisions based on the globally recorded share ratios of any requester (as opposed to rate-based). This shows that using share ratios is equally effective at achieving fairness as the traditional approach of using observed upload rates.

E. Summary

It has been shown that availability and performance are very closely linked with unavailable torrents also being low performance torrents. As such, the simulations show that both vanilla BitTorrent and rate-based bartering fail to offer high performance because they fail to improve availability.

Our experiments also show the fundamental limitations of single-torrent-based seeding time extension: here the seeding time must be extended by an extreme length (x10) to achieve high availability and performance. So far, no effective incentive mechanism exists to achieve this.

Similarly, cross-torrent bartering does not offer any real solution for the availability problem due to the low probability of successfully finding peers with synchronous interests. This results in low download performance and high abortion rates.

In contrast, both eMule-like and indirect reciprocation incentives offer extremely effective mechanisms for addressing unavailability, even outperforming the costly extension of seeding times. This is because peer contribution becomes detached from time, allowing peers to claim back contributions at a later date.

However, interestingly, the eMule-like incentives improve availability without also improving performance. This is because a peer can only recoup its contributions through repeated interactions; as such, it is impossible for a peer to gain superior performance unless it re-encounters a past peer. In contrast, indirect reciprocation offers the best performance by a significant margin due to its ability to incentivise peers to make contributions to any and all torrents, confident in the knowledge that this strategy can improve their own position.

Finally, BitTorrent's incentive scheme and the cross-torrent variants all have positive fairness characteristics with the exclusion of eMule-like incentives. This is because eMule-like incentives make unassured investments that may not be recouped in the future; this is an endemic problem of any local persistent history mechanisms. Therefore, in practise, long term users of eMule (and its variants) are likely to reduce their sharing thereby undermining its previously identified benefits.

In summary, we conclude that cross-torrent indirect reciprocation outperforms other approaches regarding the combination of all three metrics: availability, performance, and fairness.

VI. Related Work

Surprisingly, little research work has been performed into addressing the file availability problem in BitTorrent [4]–[7]. The most recent work improves file availability in BitTorrent by file bundling [4]. This solution basically aims to enlarge online times of the users and belongs therefore to the category of *single-torrent approaches*. Using a queuing theoretic model and controlled experiments on PlanetLab, the authors show that this approach can reduce waiting-time for peer in torrents with highly unavailable seeders. Our evaluation has shown, however, that such single-torrent approaches must dramatically increase seeding times (x10) to achieve 99% availability; a requirement which is often infeasible even in the presence of strong incentives.

Guo et al. [6] were the first to propose intriguing ideas and results for cross-torrent collaboration. Amongst other things, the authors sketch a very abstract mechanism for instant inter-torrent collaboration that does not consider history information. Yang et al. [5] propose a variation of these ideas by designing a cross-torrent tit-for-tat strategy that assumes repeated interactions of the users. Piatek et al. [7] propose an alternative protocol that enables long-term incentives in BitTorrent with the aid of one-hop intermediaries. Using measurement data from real BitTorrent networks, the authors impressively show that it is possible to establish a shared contribution history among nodes without inducing a significant overhead. This mechanism is therefore a prominent example of the *indirect reciprocation* scheme evaluated previously. However, the authors do not explore the resulting performance, availability and fairness aspects that arise through cross-torrent collaboration.

In summary, the above works in the area of cross-torrent collaboration differ in the way how they incite users to contribute resources as seeds in a multi-torrent environment. Specifically, contribution incentives are either built on *private* histories and direct observations [5] or *shared* contribution histories established either due to the information exchange between the nodes [6], [7]. This work has covered each of these situations with private histories being exemplified by eMule and shared histories exemplified through indirect reciprocation.

Finally, it can be observed that there are a number of private BitTorrent sites that require (invited) users to login using persistent accounts. These sites monitor how much each user uploads and downloads so to enforce a specific contribution ratio. Although this simple and non-technical solution belongs to the category of indirect reciprocation schemes, it is inherently vulnerable to misreporting [21]; for instance, there are many client add-ons that allow users to lie about their contributions, easily enabling free-riding [22].

VII. Conclusions

This paper has investigated BitTorrent's seeder promotion problem in the wild and explored the potential solution-space. To achieve this, two large-scale measurements studies were performed to (i) understand seeders in BitTorrent and (ii) to obtain sufficient data to enable accurate trace-driven simulations.

Our measurement data highlighted the seriousness of the problem: more than 38% of torrents lose their seeders within the first month and most of them only after 5 days. Once in seedless state, the download rate in such torrents quickly drops to 0 KBps and user subsequently abort downloads resulting in a chain effect that leads to future download failures.

To overcome this problem, four different approaches are considered and evaluated through extensive trace-based simulations. Most notably, it was found that

- cross-torrent bartering relying on instant incentives and direct observations fails to address the problem,
- eMule-like incentives based on local persistent history mechanisms improve availability but do so without improving performance,
- indirect reciprocation provides superior performance while maintaining availability and fairness.

Through the use of up-to-date trace-based simulations, these findings confidently show that cross-torrent collaboration is a viable solution and that indirect reciprocation offers the most promising research direction.

Acknowledgements

This work has partially been supported by the German Federal Ministry of Education and Research (BMBF) in the project Premium Services (01IA08003A).

References

[1] B. Cohen, "Incentives build robustness in bittorrent," in *1st Workshop on Economics of Peer-to-Peer Systems*, 2003.

[2] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, analysis, and modeling of bittorrent-like systems," in *IMC*, 2005.

[3] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, "Bittorrent is an auction: Analyzing and improving bittorrents incentives," in *SIGCOMM*, 2008.

[4] D. S. Menasche, A. Rocha, B. Li, D. Towsley, and A. Venkataramani, "Content availability and bundling in swarming systems," in *CoNEXT*, 2009.

[5] L. G. Yan Yang, Alix L.H. Chow, "Multi-torrent: a performance study," in *MASCOTS*, 2008.

[6] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, analysis, and modeling of bittorrent-like systems," *IEEE Journal on Selected Areas in Communications*, vol. 25, Issue: 1, 2007.

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

[8] Mininova, BitTorrent index site, http://www.mininova.org.

[9] Azureus, http://sourceforge.net/projects/azureus.

[10] Emulab – Network Emulation Testbed, https://www.emulab.net.

[11] M. Izal, G. Uroy-Keller, E. Biersack, P. A. Felber, A. A. Hamra, and L. Garces-Erice, "Dissecting bittorrent: Five months in torrent's lifetime," in *PAM*, 2004.

[12] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and improving a bittorrent networks performance mechanisms," in *INFOCOM*, 2006.

[13] The eMule project, http://www.emule-project.net.

[14] V. Vishnumurthy, S. Chandrakumar, S. Ch, and E. G. Sirer, "Karma: A secure economic framework for peer-to-peer resource sharing," in *P2P-Econ*, 2003.

[15] Q. Lian, Y. Peng, M. Yang, Z. Zhang, Y. Dai, and X. Li, "Robust incentives via multi-level tit-for-tat," in *IPTPS*, 2006.

[16] T. Bocek, Y. El-khatib, F. V. Hecht, D. Hausheer, and B. Stiller, "Compactpsh: An efficient transitive tft incentive scheme for peer-to-peer networks," in *LCN*, 2009.

[17] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW*, 2003.

[18] Maxmind, Free Geolite Database, http://www.maxmind.com/app/geolitecountry.

[19] Ookla's Speedtest Throughput Measures.

[20] A. Legout, G. Urvoy-Keller, and P. Michiardi, "Rarest first and choke algorithms are enough," in *IMC*, 2006.

[21] Private Tracker Ratio Hacking - Busted Myth? Or Reality?, http://filesharefreak.com/2008/08/11/private-tracker-ratio-hacking-busted-myth-or-reality/.

[22] Sb-innovation leeching mods, http://www.sb-innovation.de/.