Working Paper #06-14

September 2006

On the Impact of Practical P2P Incentive Mechanisms on User Behavior

Kostas G. Anagnostakis, Institute for Infocomm Research
Fotios C. Harmantzis, Stevens Institute of Technology
Sotiris Ioannidis, Stevens Institute of Technology
Manaf Zghaibeh, Stevens Institute of Technology

On the Impact of Practical P2P Incentive Mechanisms on User Behavior*

Kostas G. Anagnostakis
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
Email: kostas@i2r.a-star.edu.sg

Fotios C. Harmantzis
Telecommunications Management Department
Stevens Institute of Technology
Hoboken, New Jersey 07030
Email: fharmant@stevens.edu

Sotiris Ioannidis
Computer Science Department
Stevens Institute of Technology
Hoboken, New Jersey 07030
Email: si@cs.stevens.edu

Manaf Zghaibeh
Telecommunications Management Department
Stevens Institute of Technology
Hoboken, New Jersey 07030
Email: mzghaibe@stevens.edu

*Financial support from the NET Institute, www.NETinst.org is gratefully acknowledged.

Abstract

In this paper we report on the results of a large-scale measurement study of two popular peer-to-peer systems, namely BitTorrent and eMule, that use practical and lightweight incentive mechanisms to encourage cooperation between users. We focus on identifying the strategic behavior of users in response to those incentive mechanisms.

Our results illustrate a gap between what system designers and researchers expect from users in reaction to an incentive mechanism, and how users react to those incentives. In particular, we observe that the majority of BitTorrent users appear to cooperate well, despite the existence of known ways to tamper with the incentive mechanism, users engaging in behavior that could be regarded as cheating comprised only around 10% of BitTorrent's population. That is, although we know that users can easily cheat, they actually do not currently appear to cheat at a large enough scale.

In the eMule system, we identify several distinct classes of users based on their behavior. A large fraction of users appears to perceive cooperation as a good strategy, and openly share all the files they obtained. Other users engage in more subtle strategic choices, by actively optimizing the number and types of files they share in order to improve their standing in eMule's waiting queues; they tend to remove files for which downloading is complete and keep a limited total volume of files shared.

1 Introduction

Cooperative distributed systems, commonly known as “peer-to-peer systems” refer to a class of systems and applications that employ distributed resources to perform critical functions in a decentralized manner. The users of the system or the *peers*, are anonymous entities, sharing their resources to benefit each other within a scalable, stable and reliable global service.

P2P systems have the potential of providing a wide range of services such as file sharing, content distribution and distributed data processing. They are often contrasted with client-server systems, because participants typically blur the distinction between client and server by acting as both consumers of the service (e.g., clients) as well as producers (e.g., servers).

These systems enable massive resource and information pooling at low cost per participant and at scales that are difficult to achieve with traditional client-server systems, while local autonomy and network effects provide resilience against failures and attacks.

One distinguishing characteristic of P2P systems compared to more traditional client-server designs is the widespread cooperation between participants by sharing resources and information. As all the benefits of these systems are deeply rooted in cooperation, they are inherently vulnerable to large-scale non-cooperative behavior. It is therefore necessary for the system to be designed so that participants generally cooperate. The mechanisms that are embedded in the system for this purpose are called *incentive mechanisms*.

Economical and incentive aspects in P2P networks have been in the enhancement and the development loop since the emergence of this class of systems. Theoretically, and to fully utilize the resources of a P2P network, the system architect must have a precise understanding of the payoff of each individual user joining the network in order to construct an appropriate incentive mechanism [3]. However in practice, the architect is not able to have a clear vision of the requirements of every user. Even if he did, it is difficult to implement the incentives that would actually work. The difference between what researchers designed, and how users actually react to the proposed incentives is dependent on many factors. First, users in a P2P network perceive the proposed incentives in different ways depending on their own utility functions. Second, most of the proposed incentive mechanisms do not adapt to changes in users' behavior. Third, there is no extensive and reliable measurement study that would help P2P network designers evaluate the *real* performance of their incentive mechanisms with the *expected* performance. Most of the measurement studies completed on existing P2P networks demonstrate the success or the failure of such systems with no analysis or explanation on why these systems succeeded or failed.

In general, a failure of a P2P system is related to the lack of incentive acceptance among users, who disregard the system if they sense that it does not provide them with satisfactory payoffs. For example, some of the *payment-based* systems, such as the *Mojonation* system [22], have lost ground to other known P2P systems like BitTorrent and eMule. One of the reasons of this failure was the limitation of the payment-based approach where all transactions performed in the system had to be cleared by a centralized entity.

Furthermore, when researchers designate a P2P system as successful, they usually relate the success to the system public acceptance and approval without providing detailed evidence supporting the evaluation. That is, there is no proof that these systems are successful because they have strong incentive mechanisms, robust against security attacks, or some other reasons. Moreover, some of these systems have not been thoroughly tested against security attacks for different reasons. First, there is a vague understanding of what strategies the attackers can possibly follow. Second, users might not attempt to subvert the incentive mechanism because they already get a satisfactory level of performance even though the incentive mechanism of the P2P system is vulnerable to attacks.

In this paper, we focus on the economical and incentive aspect of P2P systems. We intend to analyze and investigate the behavior of users in two major P2P systems: BitTorrent and eMule.

Both of BitTorrent and eMule have incentive mechanisms embedded in their designs. For each of these systems, there is some utility function that represents the expected users' behavior in that system. That is, the incentives in that system yield perfect behaviors on the part of the users. However, if in practice users do not behave perfectly as expected, then it must be that their utility function does not reflect what it has been set for. So, another way of viewing our investigation is an attempt to reverse-engineer the *real* utility function by looking at users behaviors under different sets of incentives.

We present the results of our large-scale measurement study of BitTorrent and eMule. Our paper is an analysis of how users recognize the incentives provided in both of BitTorrent and eMule. Moreover, we question how users respond to the incentives, and whether the incentives provided in both systems are sufficient enough to drive users towards the desired level of cooperation.

The rest of the paper is organized as follows: In Section 2 we provide a background on both of the BitTorrent and eMule incentive mechanisms. In Section 3 we present our experimental analysis. Results and the discussion are in Section 4. Related work is in Section 5, while in Section 6 we conclude our paper.

2 Background

2.1 BitTorrent

BitTorrent [7] is a content distribution system in which peers cooperate in downloading files. It relies on lowering the cost of sharing by redistributing the load from one peer to multiple peers. Each peer can be connected to several others to download parts of the file. However, although a peer might be connected to many other peers, it typically uploads only to a small number of them. A peer decides which users to serve based on the following three rules. First, a remote peer is *unchoked* based on his current upload rate – if he provides reciprocal service, he will be given service in return. Second, a peer is *optimistically unchoked* even if he does not provide reciprocal service, in an attempt to provoke him to cooperate. Third, peers that are currently unchoked but do not provide any useful services are periodically *choked*, assuming that they are uncooperative or do not have any useful parts of the file to offer (this is called *anti-snubbing*).

The BitTorrent design shows that the mechanism differentiates between cooperative and non-cooperative peers by giving higher service rates to the cooperative ones. It is not clear, however, whether the mechanism provides a *strong enough* incentive for peers to cooperate, whether peers respond to this incentive, and how it performs compared to other mechanisms. There are informal observations of good performance (compared to other file sharing systems). This may be caused by the incentive scheme, but it may also be the result of the particular community of peers that have adopted this system. Other than informal observations, we are not aware of any firm evidence in terms of analytical results or empirical observations.

An efficiency-related issue arises because of the role of *seeding peers*. Seeding peers are peers that only upload contents. They are either special peers that are set up by the content provider to bootstrap content distribution or normal peers that have completed the download but remain in the system and provide contents to others without expecting anything in return. Naturally, peers have no incentive to remain online as seeders after they are done downloading. If the content provider does not provide seeders of his own, the distribution of a file may no longer be possible. Well-known BitTorrent sites often employ banners kindly asking users to stay online, if possible, as seeds, in order to preserve torrents. Other sites periodically compute the upload/download ratio for each user. Once a user's ratio falls below a certain level, the site bans that user until he rectifies his position by increasing his ratio.

One implication of having seeding peers in the system is that they *dilute* the incentive to cooperate. As the fraction of seeders in the system increases, a larger fraction of the total content distributed will not be subject to the tit-for-tat exchange mechanism.

BitTorrent's incentive mechanism is fully decentralized and is thus, in principle, less vulnerable to infrastructure attacks, unlike monetary mechanisms. However, it is not clear whether it is robust enough against selfish and malicious nodes manipulating the mechanism. There are several potential problems that are not addressed by the design as described in [9]:

Vulnerability to Sybil attacks: A peer can increase his download rate by presenting multiple identities to other peers because this will proportionally increase their probability of being optimistically unchoked, even if they are later choked as part of the anti-snubbing policy. Practically speaking, we do not believe this to be a major threat to BitTorrent, as long as identities are bound to IPv4 addresses, and the number of addresses and thus identities that a typical user can acquire in the current Internet is rather small.

Malicious peers can fool the mechanism by serving false content: Specifically, the mechanism will respond to incoming false content by unchoking the malicious peer and thus increasing his utility.

Parameter tweaking: Several operational parameters of a BitTorrent client can be tweaked to gain unfair advantage over other users. BitTorrent clients obtain information about other peers in the system by periodically querying a centralized *tracker*. In the default configuration, peers re-query the tracker every 5 minutes if the number of active downloads falls below a threshold of 4 connections. It is very easy (e.g., via command-line parameters) to change this behavior. Thus, if a freerider can obtain 20% of the download speed of a cooperating peer, he only needs to multiply the number of active connections by five to reach 100%, while also querying the tracker at a much faster rate in order to learn about other peers more frequently.

We ask whether this is a fundamental flaw or whether it is something that can be easily fixed. Trackers could try to enforce a limit on the number of re-queries. However, peers could collude to circumvent this measure by directly exchanging peer information between them. This kind of coordination could easily be implemented in one of the clients available for BitTorrent, and developers have a clear incentive to do that as it will make their clients perform better than others. Of course, the effect of this cheat diminishes as the number of users employing it increases, as it gradually levels out the playing field.

Similarly, users can easily change the maximum number of incoming connections that are allowed. This would increase the probability of incoming connections that periodically unchoke optimistically. Peers also exchange information on which blocks they have and which blocks they want and use this information to decide whether or not it is worth keeping a connection up (because the remote peer has some useful data). A peer could therefore go one step further and *lie* about his own local state, in order to maximize the number of active incoming connections. Finally, peers can more aggressively seek out and persist in trying to connect to seeders, as they provide guaranteed free service.

2.2 eMule

A lightweight, *pair-wise* credit system is implemented in the eMule system [10]. The goal of the credit system is to reward users contributing to the network by reducing their waiting time in the upload queue. For each request in the upload queue the peer computes the *Queue Rank* based on a scoring function that depends on the current waiting time for the request, as well the upload and download volumes for the peer. The main advantage of this scheme is simplicity: there is no communication overhead and a peer only needs to maintain the upload and download volumes for each peer it has communicated with. The approach is cheat-proof in the sense that peers have no reason to tamper with the credit file. However, anecdotal evidence [25] suggests that the approach does not consistently provide a clear performance advantage to users who contribute resources to the network. Although there is no clear evidence in terms of measurements to support a particular conjecture that explains why this is happening, the credit approach appears to have two main limitations.

First, it is hard for a peer to strategize in terms of what peers he wants to earn credit from in order to maximize expected benefits. A large fraction of peers may be disconnected resulting in delays in rewarding credit; other peers may leave the system permanently, resulting in loss of credit; others may not have any object the peer is interested in, and some may not share content at all. The use of “waiting time” as a factor in computing queue rank further complicates this problem. It results in giving a weaker performance advantage to users with established credit, as peers that do not have any credit can still use the system if they are patient enough. Tuning the scoring function to reduce the effect of waiting time is possible, but results in never serving users that don’t have established credit, even if establishing credit with those peers could be beneficial in the future.

One practical workaround to address this problem[1] is to control the set of shared files in a way that increases credit with peers likely to be useful for a given set of requests in the near future. For instance, if a peer is requesting an object in category C, then it makes sense to limit sharing to only those objects that are already available and belong to category C. Assuming that remote peers sharing the requested object are likely to request objects from the same category, the peer is more likely to earn credit and therefore improve queue rank and reduce waiting time on those peers. In this scenario, the credit system essentially *approximates* exchanges, at the cost of additional effort to get the conditions right for this to happen.

A second problem with the pairwise credit system is that there is no clear incentive for individual peers to cooperate in supporting the credit system, although this approach could in principle lead to a better global operating point. There is also no strong individual incentive not to honor credit, but in practice certain variants of the eMule client do not support the credit system, which also means that a fraction of the credit earned essentially gets lost. The mechanism does not directly penalize clients for this type of defection, and building additional protection (e.g., monitoring compliance and maintaining blacklists) adds complexity.

3 Related Work

Cooperation is highly required in P2P networks in order to entirely comprehend their potential. Indeed, if too many peers in a given P2P network choose not to cooperate, the system will reach a critical state where its utility drastically diminishes. Therefore, designers of P2P networks recount the eventual success of their networks on providing incentives to users and motivating them to cooperate.

Most of the proposed incentive techniques pivot on the concepts of rewarding peers who cooperate and punishing those who choose not to cooperate. However, they differ on how they provide the incentives to the users. Some of the adopted incentive techniques rely on reputation principles. For example, peers refer to *private history* tables, in which they save their previous interactions with other peers, to decide on the level of cooperation they will have with them in the future. On a more complicated scheme, peers collaborate to compute global trust attributes, *shared history*, by propagating their assessments of interacting with others in the network [19]. However, both of the mentioned reputation schemes have drawbacks: The private history

[1] This has been suggested on message boards as a strategy that has worked in practice.

technique does not scale to the increase in users' population, while the shared history technique is vulnerable to collusion [12].

Other types of techniques in providing incentives are the ones that adapt to changes in peers' behaviors. Such techniques are based on serving peers who can provide simultaneous and symmetric services with higher priorities [1], or by employing bartering principle as a method to negotiate rates of basic resources [8].

Reciprocity techniques have also been used in P2P economics. Tit-for-tat [4] is known to be an effective and robust mechanism that proved to thwart defection attacks. The multi-round punishing and rewarding game is meant to motivate peers to cooperate. A defective peer in the first round will be punished in the next round. On the other hand, a cooperative peer in the first round will be rewarded in the second one. However, the punishment and the reward will not be perpetual: The game will be played over again with no harboring.

Reciprocity-based protocols such as BitTorrent have been discussed in some publications. [11] argues that the tit-for-tat mechanism employed in BitTorrent has helped increasing the cooperation level among peers. Similarly, [2] emphasizes that BitTorrent's reciprocity algorithm makes the system appealing to be used and that the design of BitTorrent increases cooperation among peers. The paper also suggests that in some cases and due to the large number of seeders, BitTorrent fails in reducing freeriding since there is no specific mechanism embarked to limit their gains. On the other hand, [6] suggests that the tit-for-tat mechanism is not efficient enough in deterring unfairness relating this inadequacy to the heterogeneity of peers' bandwidths. Moreover, [18] argues that BitTorrent lacks fairness: It does not punish freeriders effectively neither it does reward users who contribute properly.

The unchoke mechanism is also discussed in some of the literatures. [9], a paper by the creator of BitTorrent, argues that the unchoke mechanism is the only efficient method used in BitTorrent to maximize peers' download rates. [21] suggests that the optimistic unchoke mechanism fortifies the system's robustness by giving leechers chances to connect to other fast leechers or seeders. However, although this study explores the robustness of BitTorrent against attacks, it does not consider the counter effects of the optimistic unchoke mechanism. Another paper that studied BitTorrent's incentive mechanism is presented in [24]. It provides an analytical representation to a simple fluid model for a BitTorrent-like system. The study briefly discusses the relation between freeriding problem and BitTorrent's optimistic unchoke mechanism. However, the limitation of this study is that it assumes a global knowledge of all peers upon selection. On the other hand,

[20] evaluates the properties of the unchoke mechanism in BitTorrent. The paper questions the behavior of the mechanism in providing reasonable reciprocation in balancing upload and download rates and how it performs with freeriders. The study provides a detailed analysis of the results obtained from experimenting the system under several conditions. The authors conclude that the unchoke mechanism provides fairness to leechers in connecting to others and a reasonable level of reciprocation in general.

A design related study is presented in [15]. The paper claims that the lack of the meta-data search in BitTorrent, which is usually regarded as a weakness in its design, might be the main reason that BitTorrent is considered as a high cooperative system. The paper argues that this design-related matter causes peers in BitTorrent to be clustered based on what fit their goals, and that clusters joined mostly by freeriders generally tend to die.

BitTorrent's threat scenarios have been mentioned in [21]. The study presents three exploits, which could possibly take place in the system, and evaluates their impact on its performance. The first threat discussed is when a selfish peer repeatedly requests lists of available peers from the tracker and chooses to download from seeders while refusing to connect to leechers. The second threat is when a selfish peer connects only to fast peers, thus, not only reducing the chances of other *slower peers* in connecting to the fast peers, but also discriminating against the slower peers by refusing to connect to them. Finally, the third threat discussed is the advertisement of false pieces.

The majority of BitTorrent's studies have been done on its performance. [6] presents a simulation based study about its performance focusing on link utilization and download time. The study confirms that BitTorrent has a near-optimally performance regarding these two metrics.

[23] studied the system's performance characteristics such as availability, integrity, flashcrowds handling and download speeds. The paper suggests modifications on BitTorrent's architecture to provide incentives for peers who become seeders to stay connected for longer times. However, the paper only advises to provide seeders with the ability to choose the bandwidth in which they want to seed.

[14] performed an extensive trace analysis on BitTorrent-like system. The study finds that the availability service in BitTorrent is poor and does not attain a satisfactory level. Furthermore, the study relates the existing problems, such as the fluctuating in downloading performance and exhibiting unfairness to peers, to the exponentially decreasing arrival rate of peers.

On the other hand, [17] traced a torrent for five-month period in which thousands of peers were involved.

The paper concludes that BitTorrent is an effective system which allows peers to obtain high download rates comparing to other P2P systems, and that it reacts robustly towards flashcrowds. However, the paper does not go further to experiment the efficiency of the incentive mechanisms in BitTorrent. Likewise, [5] concludes similar outcomes, however, it traced larger number of contents that varied in size and popularity. Furthermore, it explored the side of files' disseminating and load balancing in BitTorrent.

On eMule side, and to the best of our knowledge, there is no direct detailed study that involves measuring or evaluating its performance. However, we found a workload measurement on eDonkey which is an eMule-like system. The measurement traced eDonkey's clients in order to get information about their shared contents. The distribution of the shared contents in that study shows that there is a degree of geographical and interest-based clustering in the eDonkey system. The paper also suggests that this type of clustering could be exploited in enhancing search mechanisms for such systems.

An incremental work on [13] is presented in [16]. The work was also conducted on eDonkey in order to evaluate the semantic relationships observed among peers. The paper questions how to capture and exploit the semantic relationships among peers without involving them directly.

The impact of *guarded hosts* on the performance of eDonkey file sharing system is discussed in [26]. Guarded hosts are end-hosts who reside behind Network Address Translator gateways or firewalls. The study used a modified eMule client to collect list of peers who are connected to servers. The client employed in the study requested the lists of peers who carry certain types of files chosen based on their availability. The study found that about 23-36% of eDonkey peers were located on guarded hosts. Furthermore, the paper suggests that the existence of guarded hosts inhibits fair sharing among peers and that it is important to consider the employment of guarded hosts when designing or improving a peer to peer system.

4 Methodology

A BitTorrent peer normally reports several attributes to the network such as its IP address, downloaded volume, and its connection time. However, the peer does not report its uploaded volume which is an essential attribute to get a complete view of the cooperation level.

Our BitTorrent data collection we utilized the following methods: (i) A modified BitTorrent client that is able to aggressively request new peers from the main tracker and try to connect to as many peers as possible to track their download progress and their connection time. The tool is developed using a robust C-language BitTorrent client called *ctorrent*. We modified the ctorrent client into *ctorrent-bigbro* to be able to only connect to the remote peers and monitor their progress without being involved in any download or upload activity, to avoid accidentally accessing illegal content. The ctorrent-bigbro provided us with remote peer's download volumes and their connection times only. The remaining attribute, which is the uploaded volume, was calculated from the IP-ID field which is embedded within each packet sent by remote peers.

(ii) The other method we used for getting detailed information about the performance of BitTorrent peers was obtaining the information directly from trackers. Trackers provided us with full logs of each peer's activity, including the download/upload volume and the connection time.

In our data collection process on eMule we used a client stripped from the original eMule client's GUI. The client is called *xmule-crawler*. xmule-crawler is capable of collecting detailed information that is not provided by using other passive monitoring techniques. Such information include requests issued by the user, aborted requests, download progress and files being deleted or removed from the shared directory.

On the BitTorrent side, we traced around 1000 torrents, involving more than a hundred thousand peers. On eMule, we obtained a total population of about 3000 peers with around ten days worth of tracking for each peer.

5 Results and discussion

We analyze our data in an attempt to understand how users in BitTorrent and eMule systems perceive and respond to the incentives provided; and whether the incentives are strong enough to drive users to a satisfactory level of cooperation.

In BitTorrent, we investigate how users follow different strategies by monitoring their download and upload behaviors and in that fashion deducing their cooperation levels. The strategies are set by fine-tuning the parameters of the BitTorrent client. Some of these parameter changes could boost the download progress of users and advance their gain compared to others who are probably not aware of the existence of these options or their significance. Such behavior induces freeriding. Currently, there is no clear understanding about the magnitude of freeriding in BitTorrent. Most of the work on BitTorrent is likely to focus on the optimistic unchoke mechanism and particularly whether it is an effective technique to deter freeriding.

Although the optimistic unchoke process is used to incite peers who do not provide reciprocal services to cooperate, we suspect that it also furnishes opportunities for peers to freeride.

Similarly, in eMule, we investigate how effective the pairwise credit mechanism is in motivating users to cooperate. In general, the credit mechanism in eMule implies that sharing is the sole factor affecting users' positions in the upload queues and therefore rewarding them by reducing their wait-times. In practice, users may not follow this basic strategy. We suspect that they may, instead, tend to develop their strategies by undertaking different sets of *sharing actions* based on what fits their interests in downloading and uploading.

For an eMule peer, the shared directory is the only accessible directory to the rest of the network in which the peer saves the content he wishes to share with others. Simply put, the shared directory is the place where an eMule client saves the ongoing downloads. If a peer does not want to share a content, he can simply remove it from the shared directory and place it in another inaccessible directory. Therefore, by constantly following changes in the sizes of the files in the shared directories, we can realize peers' cooperation levels. Moreover, by tracking how often users remove the completed files we shall be able to deduce their sharing strategies.

5.1 BitTorrent

As mentioned in Section 2, BitTorrent uses the unchoke mechanism to reciprocate services to nodes that cooperate and periodically choke nodes that choose not to cooperate. It also employs the tit-for-tat strategy that is known to provide a strong incentive for cooperation.

After weeding out the uninformative samples from our data, we ended up with 700 samples that showed distinctive trends in users' download/upload speeds. In each of these remaining samples, we isolated four different clusters of BitTorrent users. Each cluster is characterized by different level of cooperation exhibited by its members.

The first cluster we isolated includes users who profited from the system as much as they benefited it: The volume downloaded by those users almost equals the volume they uploaded. Contemplating in this behavior, it seems that the members of this cluster, who comprised the majority of users in our samples, did not follow any definite approach in downloading or uploading their files. Or in other words, they just simply ran the BitTorrent client on its default without changing any of its settings.

The second cluster we isolated in our samples contains peers who we classify as the major contributor leechers in the system. Their high cooperation level is clearly seen in the considerable volume they uploaded which is more than twice than the volume they downloaded. Based on that behavior, it seems that the members in this cluster are either altruistic or, as the members in the first cluster, they did not follow specific strategy in their interaction with other peers. However, they are differentiated from the peers in the first cluster by their higher bandwidth capacities. Nonetheless, we suggest that members in clusters 1 and 2 are the honest peers who responded to the incentives provided since they exhibited medium to high levels of cooperation. However, it is not clear whether their reaction to the incentives was driven by altruism, carelessness, or by the lack of knowledge of the possibility in exploiting the resources of the system and freeriding.

The third cluster in our samples includes freeriders: Users who completed their downloads without contributing to the system. The maximum upload speed we observe from those users is less than 2 Kbps. On the other hand, they were fast enough downloading with speeds reached up to 300 Kbps. Moreover, as an expected selfish conduct, those peers left the system as soon as they completed their downloads.

The last group contains users who downloaded at least as twice as they uploaded. They are also characterized by disconnecting immediately after they completed their downloads. They remarkably benefited from the system as well. However, their limited contribution to the system makes them distinguishable from freeriders.

Obviously seen, peers in cluster 2 are structuring an enjoyable downloading source for freeriders and peers in the fourth cluster. The behavior of peers in the fourth cluster is explicable if we assume that they follow advanced sharing strategies comparing to peers in clusters 1 and 2. Those peers look like they are fairly acquainted to BitTorrent's mechanisms, or at least, aware of the parameters put available in most of BitTorrent's GUI clients.

We suspect that parameters tweaking in BitTorrent counter affects the optimistic unchoke mechanism. In general, using such parameters helps users to strategize themselves to augment their gain on the account of honest peers. Nevertheless, since the population of such peers is so far limited in BitTorrent, the system is still capable of performing significantly and is still an attractive choice for users. However, if the population of such peers tremendously increased, we believe that the optimistic unchoke would be incompetent in handling such scenario.

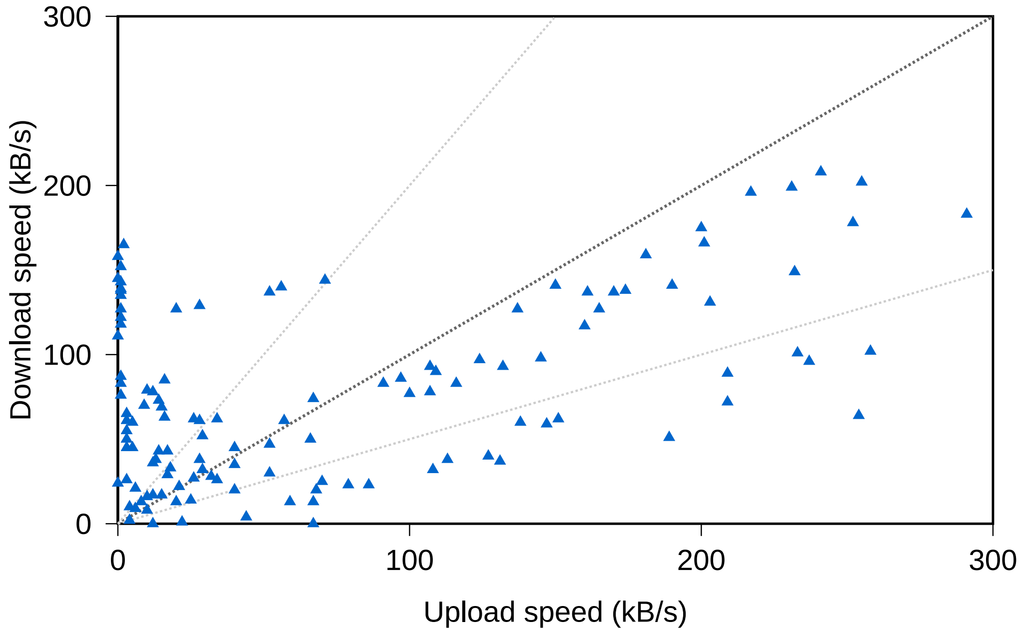

Figure 1: Upload speed vs. download speed for a sample torrent.

On the other hand, we suggest that the freeriders' behavior is either related to even more advanced strategies comparing to other peers, or to cheating. However, since the upload volumes of such peers is extremely limited, we are inclined to believe that this behavior is closer to cheating than parameters tweaking. With some modifications to a BitTorrent client, users will be able to apply several threat scenarios that allow them to completely freeride without contributing to the system. Some threat scenarios are very well known now: Serving false pieces, connecting directly to seeders, and using multiple identities.

Fig. 1 shows the distribution of the download and upload speeds of peers joined in a torrent. Freeriders are positioned on the download speed axis. Users in cluster 4 are located between the download speed axis and the $y = 2x$ line. The $x = 2y$ line separates users in clusters 1 and 2.

The distribution of users in our samples clearly illustrates the four clusters we discussed above. Fig. 2 shows the fraction of each cluster in each torrent in our samples. The majority of users belong to cluster 1 and cluster 2. The figure shows that at least 50% of users in all of our samples uploaded as much as they downloaded. Users in cluster 2 who have been taken advantage of formed about 17%. On the other hand,

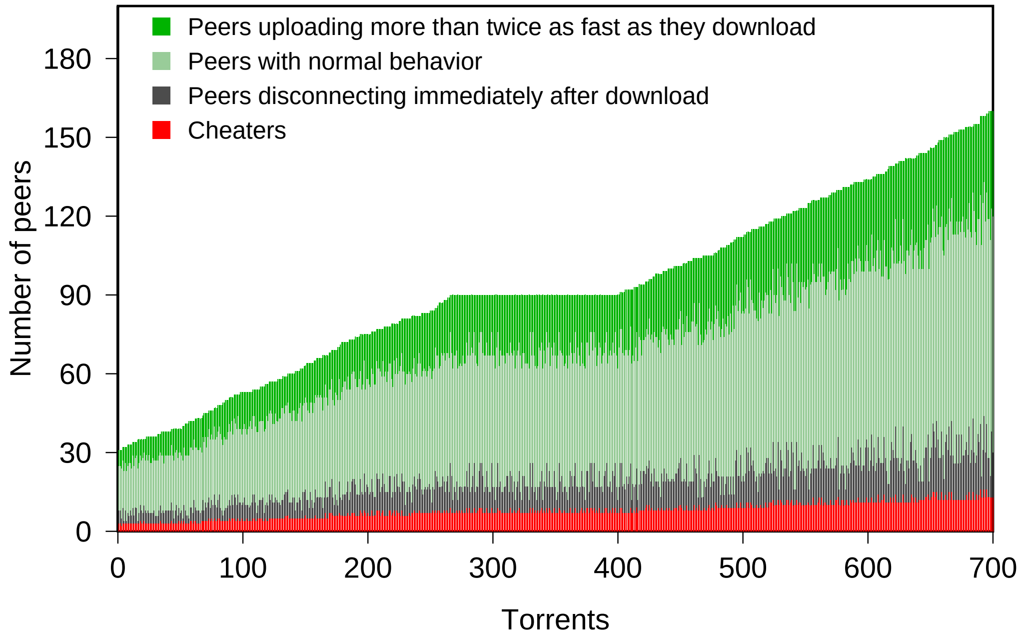

Figure 2: Distribution of peers joined in 700 torrents.

cluster 4 or users who disconnected immediately after they completed their downloads comprised about 10% of the total number of users in a torrent, while freeriders population reached about 10%.

5.2 eMule

On the other hand, the credit mechanism in eMule rewards users who allocate resources for sharing by reducing their waiting time in the upload queue and thus expediting their downloads.

In general, users realize incentives in different ways and they react to those incentives according to their comprehensions. To obtain a better understanding on how users in our eMule samples react to incentives, we have categorized them into two clusters.

The first cluster we isolated contains users who seem to perceive that sharing is the sole principal behind advancing their positions in the waiting queue. Therefore, they commit themselves to this primal action not undertaking any advanced strategy in sharing their contents. Their interaction with the system implies that they offer most of, if not all, their downloaded contents for sharing. This basic effortless behavior, which

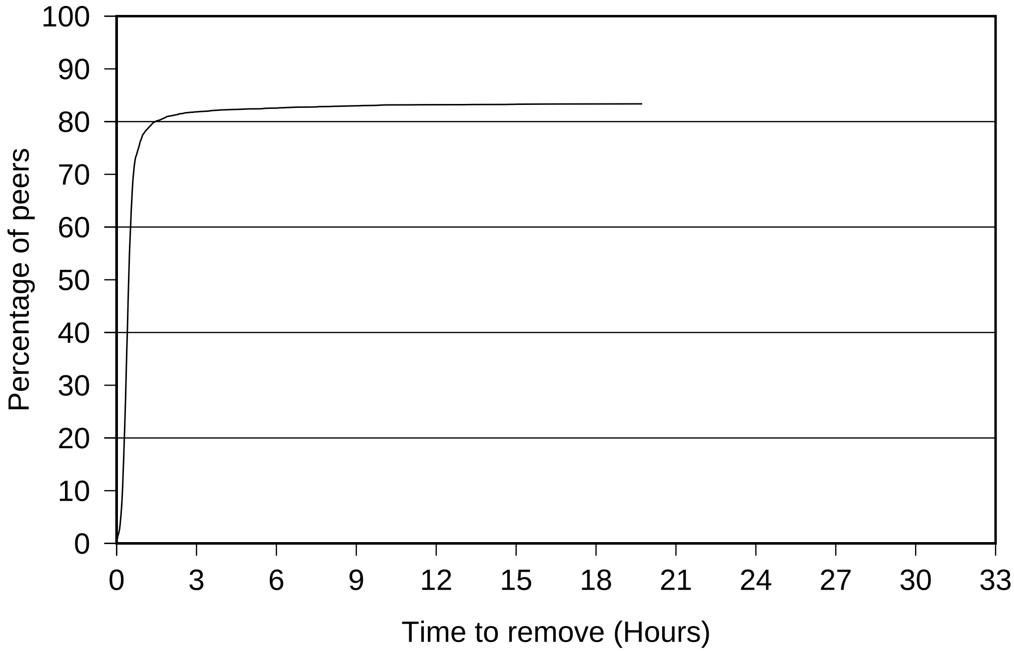

Figure 3: CDF for the time to remove files from the shared directory.

could be related to idleness, causes the shared directories of such users to grow huge. However, and to distinguish between the behavior of the users in this group and idleness, we monitored their activities during the collection of our data and we found that the majority of those users were active.

The second cluster we identified in our samples contains users who probably have a different perception of the incentives. They seem to recognize that there is no need to accumulate all the files they obtained in the shared-directories; or that there is a certain level of sharing which is sufficient enough for them to be rewarded. Thus, their behavior suggests that they advanced their sharing strategies to be based on keeping a minimum size of contents shared while removing other contents. That minimum sizes of what they shared differ among them and probably depend on personal preferences or factors such as the popularities, the sizes or the categories of the contents.

Our data illustrate that the members of this cluster frequently removed the complete downloaded contents from the shared directories. Nevertheless, the lapse of time in which they removed those contents varies: Some of the users removed the contents immediately after they downloaded them, others waited

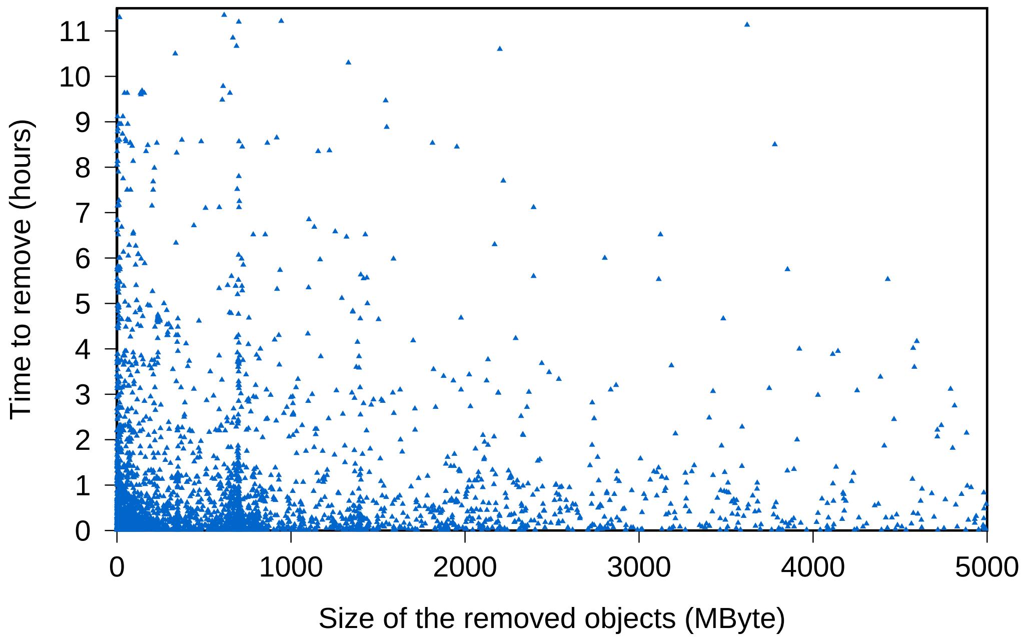

Figure 4: Time to remove vs. size of the removed object(s) for sizes less than 5000 MBytes.

longer.

The distribution of times between completing the download of contents and removing them from the shared directories gives us an insight on the volume of each cluster in our samples. Fig. 3 presents the cumulative distribution function for the times to remove the files. It shows that about 18% of the users in our sample shared all the files they downloaded without removing any of them. On the other hand, 75% of the users removed their contents from the shared directories within one hour of completion download.

The 18% of users mentioned above represents the first cluster we isolated: Peers who we assume that they did not take any specific strategy in sharing their contents. During our sampling, some of those peers shared up to 100 GBytes of contents without removing any of them.

The 75% of users signifies members in the second cluster who frequently removed contents from the shared directories. However, users delete or remove files for several reasons not only for strategizing themselves in a download queue. Therefore, in an attempt to focus on sharing and strategy related motives, we assume that any size reduction that is more than 5 GBytes is related to non-behavioral basis such as space constraint or disk cleanup.

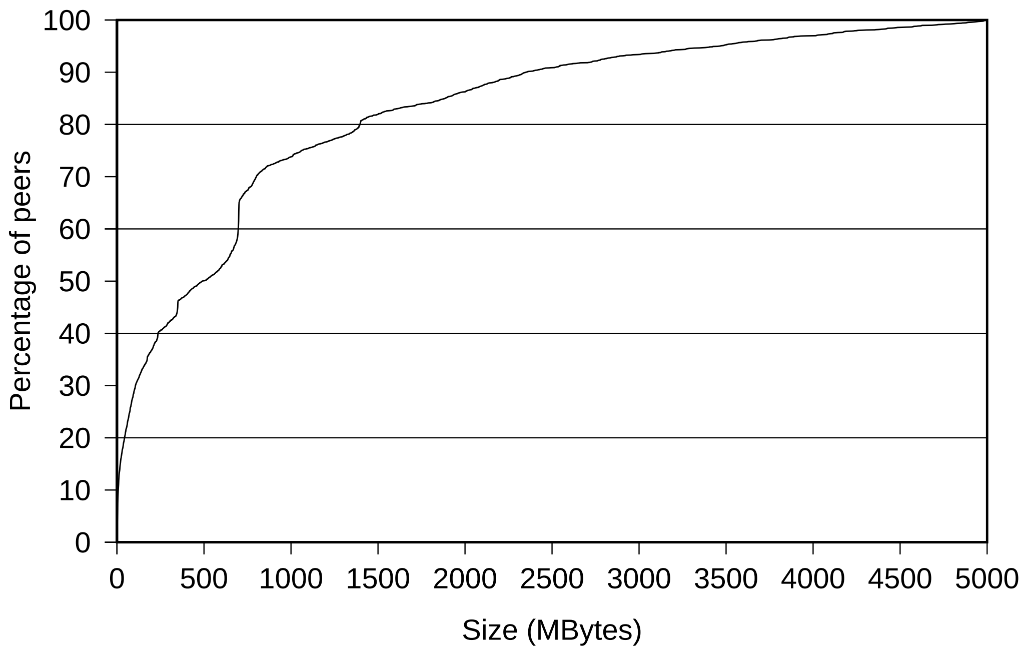

Figure 5: CDF for sizes of the removed files excluding removed files with sizes more than 5000 Mbytes.

Fig. 4 presents the distribution of the removed sizes that are less than 5 GBytes from the shared directories. The figure shows that the members of this cluster were more likely to remove small to midsized completed files. The cumulative distribution function of the sizes of the removed files, Fig. 5, demonstrates that 70% of those files ranged between 0 to 1400 Mbytes. Moreover, about 20% of them had sizes close to 700 Mbytes, which is usually a typical size of CD images, movies and other media files.

6 Conclusion

In this paper we have presented a measurement analysis study of BitTorrent and eMule P2P file sharing systems.

Our findings support that peers in BitTorrent might remarkably enjoy higher download speeds comparing to other P2P file-sharing systems. It appears that the incentive mechanisms adopted in BitTorrent are succeeding in promoting cooperation among peers until now. However, the level of cooperation in BitTorrent does not seem to be as satisfactory as expected.

Our results show that while the majority of peers in our samples contributed to the system as much as they benefited from it, 17% of the peers were the main contributors to the system, and more than 10% of peers downloaded as twice as they uploaded to others. Moreover, our results show that freeriding in BitTorrent is more widespread than it was previously assumed. The percentage of freeriders in our samples reached up to 10% of the total population joined in a torrent. This is an indication that BitTorrent may not be providing a strong enough incentive to reduce freeriding since there is no explicit mechanism in its design to punish or at least discourage freeridng.

On eMule side, it looks like peers comprehend the incentives differently. Some of the peers seem to understand that sharing is good. Therefore, they tend to share most of their downloaded contents to advance their positions in other peers' upload queues. In our samples, 18% of peers kept all the files they downloaded in their shared directories. On the other hand, another cluster of eMule peers seem to realize that they only need to share a limited amount of contents to expedite their downloads. Thus, they frequently removed the downloaded files from their shared directories. Our results show that 75% of users in our eMule samples removed their contents within one hour of completing their downloads.

References

[1] K. G. Anagnostakis and M. B. Greenwald. Exchange-based incentive mechanisms for peer-to-peer file sharing. In *Proc. 24th International Conference on Distributed Computing Systems (ICDCS 2004)*, March 2004.

[2] N. Andrade, M.Mowbray, A. Lima, G. Wagner, and M. Ripeanu. Influences on cooperation in bittorrent communities. In *3rd Workshop on Economics of P2P Systems (P2P Econ)*, August 2005.

[3] P. Antoniadis, C. Courcoubetis, and R. Mason. Comparing economic incentives in peer-to-peer networks. Technical report, Special Issue on Network Economics, Computer Networks, 46(1), 2004.

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

[5] A. Bellissimo, B. N. Levine, and P. Shenoy. Exploring the use of bittorrent as the basis for a large trace repository. Technical Report 04-41, University of Massachusetts Amherst, June 2004.

[6] A. R. Bharambe, C. Herley, and V. N. Padmanabhan. Analysing and improving bittorrent performance. Technical Report MSR-TR-2005-03, Microsoft Research, Microsoft Corporation One Microsoft Way Redmond, WA 98052, USA, February 2005.

[7] BitTorrent Web Page. http://www.bitconjurer.org/BitTorrent/.

[8] B. Chun, Y. Fu, , and A. Vahdat. Bootstrapping a distributed computational economy with peer-to-peer bartering. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[9] B. Cohen. Incentives build robustness in BitTorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[10] The EMule Project. http://www.emule-project.net/.

[11] M. Feldman and J. Chuang. Overcoming free-riding behavior in peer-to-peer systems. *ACM Sigecom Exchanges*, 6(1), July 2005.

[12] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Incentive techniques for peer-to-peer networks. In *ACM Conference on Electronic Commerce (EC'04)*, May 2004.

[13] F. L. Fessant, S. Handurukande, A.-M. Kermarrec, and L. Massoulie. Clustering in peer-to-peer file sharing workloads. In *3rd International Workshop on Peer-to-peer systems (IPTPS 04)*, February 2004.

[14] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. A performance study of bittorrent-like peer-to-peer systems. In *IEEE Journal on Selected Areas in Communications (IEEE JSAC), first Quarter of 2007*.

[15] D. Halesand and S. Patarin. How to cheat bittorrent and why nobody does. Technical Report PUBLCS-2005012, May 2005.

[16] S. Handurukande, A.-M. Kermarrec, F. L. Fessant, and L. Massoulie. Exploiting semantic clustering in the edonkey p2p network. In *SIGOPS European Workshop*, pages 109–114, September 2004.

[17] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. A. Hamra, and L. Garces-Erice. Dissecting bittorrent: Five months in a torrent's lifetime. In *Passive and Active Network Measurement 5th International Workshop (PAM 2004)*, April 2004.

[18] S. Jun and M. Ahamad. Incentives in bittorrent induce free riding. In *ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, August 2005.

[19] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the Twelfth International World Wide Web Conference*, May 2003.

[20] A. Legout, G. Urvoy-Keller, and P. Michiardi. Rarest first and choke algorithms are enough. Technical Report inria-00001111, INRIA, Sophia Antipolis, September 2006.

[21] N. Liogkas, R. Nelson, E. Kohler, and L. Zhang. Exploiting bittorrent for fun (but not profit). In *The 5th International Workshop on Peer-to-Peer Systems (IPTPS 06)*, February 2006.

[22] MojoNation Web Page. http://www.mojonation.net/Mojonation.html/.

[23] J. A. Pouwelse, P. Garbacki, D. Epema, and H. Sips. A measurment study of the bittorrent peer-to-peer file-sharing system. Technical Report PDS-2004-003, April 2004.

[24] D. Qiu and N. B. Shroff. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *ACM Sigcomm*, August 2004.

[25] The eMule Project. Official eMule-Board. http://www.emule-project.net/board/index.php.

[26] W. Wang, H. Chang, A. Zeitoun, and S. Jamin. Characterizing guarded hosts in peer-to-peer file sharing systems. In *IEEE Global Communications Conference (Globecom 2004)*, November 2004.