The Impact of Incentives on the Efficiency of BitTorrent Peer to Peer Networks

Seth Cooper, Jenny Liu
CSE 561

Abstract

Peer to peer (P2P) networks have already become a popular method of distributing resources (commonly files) across the Internet. An important aspect of P2P networks is that peers must share their own resources before another peer can access them. The simplest solution to this resource-sharing problem is to rely solely on altruistic peers to share their resources (e.g. Napster). A recently introduced P2P network protocol, BitTorrent, provides a built-in incentive for peers to share. We evaluate the effectiveness of the BitTorrent incentive by comparing the efficiency of standard BitTorrent networks with BitTorrent networks where the incentive to share has been removed. We measure efficiency from the point of view of the peer/server who is the first to share a given file (as the number of bytes that peer needs to send) and from the point of view of peers/clients attempting to access a resource (as the average time needed to download a file).

1 Introduction

We consider the impact of incentives on two measures of efficiency in the BitTorrent peer to peer network. The first measure is motivated by our view of BitTorrent as an approximation of a multicast network, in which we consider the amount of work performed by the source. The second measure considers the average download time for all clients downloading the same file, in which we consider a reasonable indicator of global efficiency.

BitTorrent Overview

A BitTorrent network is built up to help the peers connected to it share a single file. This file is split up into pieces for ease of sharing. There are several components that work together to build up a network. One of the components is known as a torrent file, or simply a torrent. The torrent is a file containing metadata about the file to be shared; generally these are things such as filename, size, and hashes of the file pieces. One crucial piece of data that the torrent contains is the address of a tracker. A tracker is a process running on a host that helps peers locate each other and to coordinate themselves. Once the tracker is running, a single altruistic peer which already has the file announces to the tracker that it is sharing the file. This peer is known as a seed. At that point, any other peer who can locate the torrent file, for instance, by searching the web, can the announce itself to the tracker. The tracker will then inform them of the other peers sharing the file so that they can be contacted and can begin sharing pieces of the file with each other.

To provide an incentive to share, each downloading peer uploads to the peers from which it receives the fastest download rate. It remains connected to the other peers, but does not send them pieces; they are considered to be "choked". Thus, it is in a peer's best interest to upload as fast as possible to other peers, so that they will upload to it. In order to test for peers who might give faster download rates that the current unchoked peers, each peer reserves a connection for uploading to a randomly selected peer for 30 seconds at a time, or "optimistic unchoking". If the optimistic unchoke reveals a peer who provides a better download rate than the peer with the slowest download rate to which this one is actively connected, the unchoked peer remains unchoked and the other become choked. Once a peer finishes downloading, it will become a seed for the network, and will upload to those who download from them the fastest until it disconnects.

The BitTorrent protocol claims to build "robustness" through incentives [4]. Pareto efficiency is a state in which no change can be made to make someone better off without making someone else worse off. Specifically, a BitTorrent network achieves pareto efficiency with respect to the resources offered by each peer and the resources desired by each peer, where the resources offered by a peer are defined as file pieces that peer has and upload bandwidth available, and the resources desired by a peer are defined as the file pieces desired by that peer subject to the download bandwidth available that peer has access to. However, not every pareto efficient outcome is necessarily desirable. If we care about global efficiency, for instance, there may be a way to make one person a little worse off and someone else a lot better off, thus increasing global happiness. As a concrete example, consider a subset of peers that have a slower connection than everyone else. It may be globally desirable to upload some amount to them at the beginning so that they can trade amongst themselves rather than wait for optimistic unchokes from everyone else. More generally, pareto efficiency does not give a very strong guarantee of global efficiency.

BitTorrent also claims to achieve "a higher level of ... resource utilization than any currently known cooperative technique." [4] However, any claims about BitTorrent's resource utilization qualities are based on unicast communication between peers, and since the same file is being traded amongst all members of a BitTorrent network, much of the data transferred on the network will be replicated, thus wasting bandwidth. This observation implies that BitTorrent resource utilization may not be occurring in the best possible way.

Given these considerations, it is natural that the efficiency of BitTorrent networks should come into question.

2 Motivation

Efficiency Questions

Incentives in the BitTorrent P2P network encourage peers to trade pieces of files that they are currently downloading. However there have been many P2P file-sharing systems in the past, such as Napster, that worked reasonably well without offering direct incentives for users to play nice. We ask the questions: Are incentives really necessary? How does BitTorrent's incentive effect the efficiency of the network? Assuming that some proportion of users on a P2P network will be greedy and the rest will contribute to the resources of the network, how does the percentage of greedy peers effect the efficiency of the network? It is possible that the incentive BitTorrent supplies may actually decrease efficiency by adding an unnecessary overhead?

Efficiency Metrics

One of the goals in having the users downloading a file trade amongst themselves is to relieve the burden on the original server for the file. In a perfectly efficient sytsem, like a perfect multicast tree, the original server will only have to send out one copy of the file. In a strictly unicast system, the server will have to send out n copies, where n is the number of clients receiving the file. Thus, it makes sense to consider comparing BitTorrent's performance to the performance of multicast.

The main motivation behind multicast is to save the server from sending out multiple copies of a file over links when it only needs to be sent once. With unicast, there is a lot of wasted bandwidth on links close to the source and a lot of excess strain on the host server. This is a strain on people who pay for their bandwidth because bandwidth is not free.

Although there is a protocol for IP level multicast that is currently deployed, it's difficult to get individual ISPs to turn it on because they have nothing to gain from being the only ISP with IP level multicast enabled, and because it can disrupt their load balancing scheme. As an alternative to deploying IP level multicast, many people have turned their attention to overlay networks, which attempt to approximate true IP level multicast by implementing a network over the current IP unicast layer [3]. A common type of overlay network used for distributing files is a peer to peer network, which is a network of decentralized hosts where peers talk directly to one another rather than going through a server.

With this in mind, we choose as one measure of efficiency the "percentage of unicast", defined as the ratio of the number of bytes the initial seed on the BitTorrent network sent out to serve all of it's clients to the number of bytes the source would need to send out to serve all of its clients over unicast. A fully unicast distribution from the source to all clients would measure in at 100% of unicast, while a fully multicast distribution from the source to n clients would be $\frac{1}{n} \cdot 100\%$ of unicast.

For the second measure of efficiency, we consider the average time it takes any peer to download a file. We measure from the time a peer first announces itself to the track to the time the peer completes the file download. We consider this to be a reasonable indicator of global efficiency.

3 Previous Work

Many researchers have previously looked at the questions raised by P2P networks. There have been studies that try to decide to what extent greediness exists on the network. For instance, [1] ran a study of the Gnutella peer to peer network and discovered that around 70% of peers were freeloading and not providing any resources to the network. Similar studies have looked at how many greedy peers exist on the network [8]. Other people have tried to come up with effective incentives to encourage peers to share. Incentives are often looked at from a game-theoretic point of view, where people try to find a system such that when peers act individually in their own best interests, they will collectively do what is in the best interest of the network as a whole [2]. One of the most interesting ideas is to apply one of the most powerful real-world incentives, currency, to P2P. Many frameworks for digital currency exist, such as those presented in [5, 10, 11].

There is work that confirms that there are many greedy peers on the network who do not share and there has been much work looking at providing peers with an effective incentive to share. To our knowledge however, there is not much work looking at how much of a problem these greedy peers actually pose for the network. The assumption that providing an incentive will make the network "work a lot better" is made [6], however, it is not at all clear exactly just what "a lot" means. Other papers make similar leaps from the existence of freeloaders to needing to find a solution [5, 10]. This made us even more curious about what the answers to our questions might be.

4 Methodology

Experimental Setup

To experiment with a BitTorrent P2P network, we used the freely available Python BitTorrent source code (`http://www.bittorrent.com`). We ran our experiments with both the unmodified (vanilla) BitTorrent source and a version that we had modified to remove the incentive, by breaking the connection between upload and download rate.

Each experiment consisted of an initial seed distributing a file to a set of peers. The same randomly generated 100MB file and corresponding torrent were used across all experiments. Hosts were shared machines on the UW CSE network that were lightly loaded; BitTorrent processes running on remote machines were given a high nice value. Experiments proceeded as follows. First, a tracker is created on the host specified by the torrent. Then the initial seed verifies its copy of the file and announces itself to the tracker. Each of the peers wanting to download the file waits some amount of delay (to be discussed later) before announcing itself to the tracker. Once a peer's delay is up, it announces itself to the tracker and begins downloading the file. The behavior of a peer during and after download is determined by its strategy. When a peer finishes downloading the file, it announces this to the tracker. Eventually, all peers will have finished downloading the file, and the tracker will bring itself down, which notifies all peers that the experiment is over. In order to stagger the starting times of the peers, we have each peer to connect to the network at a random point in time within a predetermined window. The length of this window is determined by what we call the "delay coefficient"; we determine that length by multiplying the number of peers who will connect by the delay coefficient. By multiplying by the number of hosts, the relative density of hosts downloading at some particular time would remain relatively constant, even if the number of hosts changed.

Peer Strategies

In our experiments peers have one of two strategies: they are either altruistic or greedy. In a vanilla BitTorrent network, altruistic peers upload while they are downloading. Once they are done downloading, they will continue to seed the file until all peers have finished. A greedy peer will upload as it is downloading, because it is in its best interest to upload to the peers that are downloading so that they will upload to it. However, it will stop uploading to other peers as soon as it has finished downloading the file. This is not technically the greediest a peer could be; for instance, a peer could attempt to exploit problems in the BitTorrent protocol or implementation [9] or simply complete its download by waiting for optimistic unchokes. However, the greedy strategy we use applies to the greatest level of greediness of a casual user, a kind of "greedily obedient" strategy. In our modified no incentive BitTorrent network, an altruistic peer will upload to the peers who can download from it the fastest, regardless of how fast they are downloading from that peer, and will continue to do so even after it has finished downloading, until all peers have the file. A greedy peer will only download and will not upload at all. This corresponds to a "greedily obedient" strategy on a network like Napster, where peers can easily turn off uploading entirely, but will not be penalized by the network for doing so.

During the experiments, each peer recorded information about its download, including the time it took from the initial announcement to the tracker until the download was finished. The initial seed also recorded the number of bytes it had to send out. We were then able to accumulate this data and use it to calculate average download time and percent of unicast.

Network Parameters

We then chose several different parameters of the network to vary, and observed how varying these different parameters affected the efficiency of the network based on our metrics. The four parameters we varied were:

1. The minimum number of unchokes each peer would support.
2. The delay coefficient of the peers.
3. The number of peers on the network (not including the initial seed).
4. The percentage of peers who used the greedy strategy.

The default setting for the trials was 10 peers, 50% of them greedy, a delay coefficient of 5 seconds, and 4 minimum unchokes. The percentage of greedy peers was chosen to be relatively close to the believed percentage of peers who are greedy on real networks, and also to still have meaning when there were only 2 hosts downloading. The default number of peers is reasonable for a relatively small sized BitTorrent network. The delay coefficient was chosen to be a value so that the peer connections were acceptably spread out. Four minimum unchokes is BitTorrent's default setting. We then varied each of the values while keeping the other three constant. In different trials percentage of greedy peers was assigned the values 0, 25, 50, 75, 90, and 100. Number of peers varied as 2, 4, 8, and 16; delay coefficient as 0, 2, 5, and 10; and minimum unchokes as 1, 2, 4, and 7. We chose these values to give us a reasonable coverage of a wide range of values. The minimum unchokes parameter was capped at 7 because that is BitTorrent's default value for maximum unchokes. For each setting of parameters we gathered data from three experimental runs on both vanilla and modified BitTorrent networks.

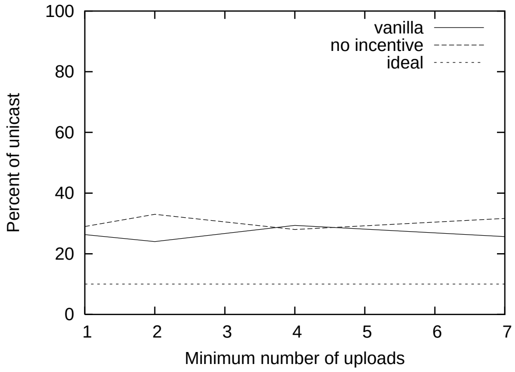

Figure 1: varying minumum number of unchokes vs percent of unicast

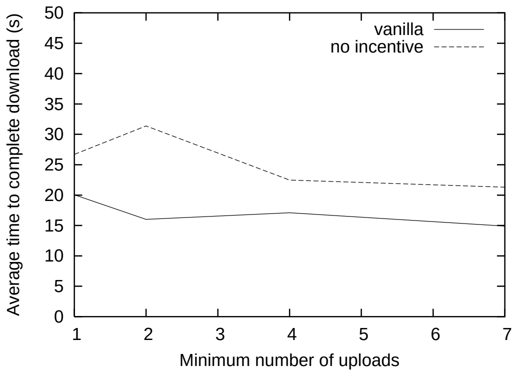

Figure 2: varying minumum number of unchokes vs average download time

5 Results

Varying the minimum number of unchokes

BitTorrent's default for minimum and maximum number of unchokes is set at 4 and 7 respectively. If the limited number of active connections that a peer had was not enough to fully utilize its upload bandwidth, increasing the minimum number of unchokes could allow peers to make more use of their upload bandwidth. We ran trials with minimum unchokes set from 1 up to 7 (all peers had the same number of unchokes set for each trial). We saw no major differences in either average download time or percentage of unicast between the BitTorrent and no

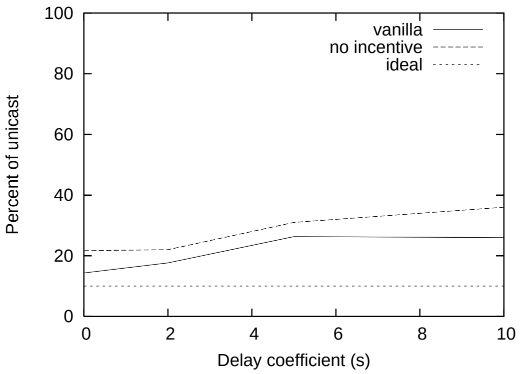

Figure 3: varying delay vs percent of unicast

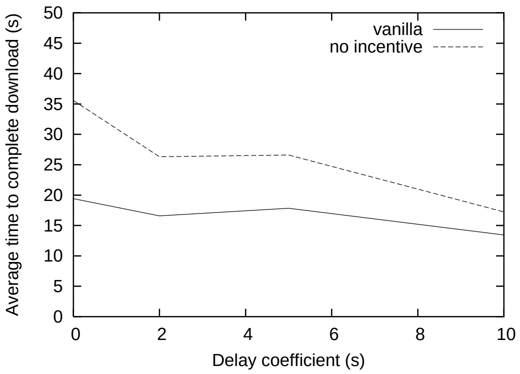

Figure 4: varying delay vs average download time

incentive networks, or across different minimum unchoke settings. This implies that the default number of unchokes was enough to make full use of each peer's upload bandwidth. However, our trials were run over a homogenous physical network and it's possible that different behavior would be observed on peer to peer networks built over heterogenous physical networks.

Varying the delay coefficient

Average download time

Increasing the delay coefficient causes the average download time to decrease in both the vanilla and no incentive versions of BitTorrent. This is probably because the likelihood that there will already be another peer on the network who has file pieces to share when a user joins the network is increased. The probability that an altruistic user will finish downloading the file and act as a seed is also increased.

In the no incentive version of BitTorrent, a greater improvement in average download time is seen as the

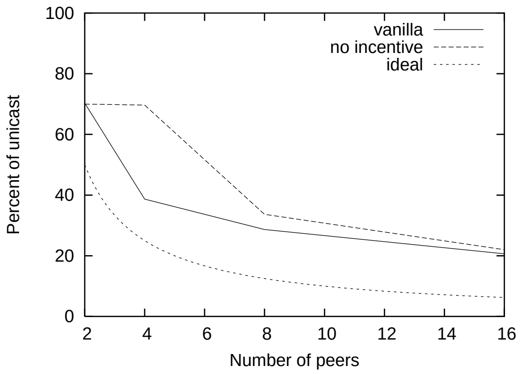

Figure 5: varying number of peers vs percent of unicast

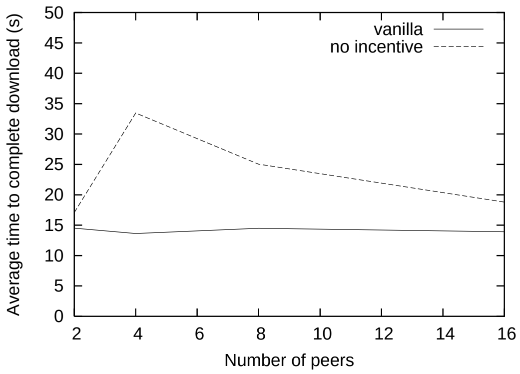

Figure 6: varying number of peers vs average download time

delay coefficient increases. This is probably because of the relatively greater gain resulting from an increased absolute number of cooperative hosts in that case (as in the case of improvement in average download time when the number of hosts increased above).

This is slightly counter-intuitive since we would expect an increased delay coefficient to contribute to content lifetime problems. However, content lifetime does not become an issue here for two reasons: 1) the delay coefficient would have to be bigger, on the order of the amount of time it takes to download the entire file, and 2) the number of users who stay on the network and continue to upload after they are done downloading the file would have to be very small (less than the 50% that was seen in this trial, as we will see later).

Percentage of unicast

The percentage of unicast increases as we increase the delay coefficient in both vanilla BitTorrent and

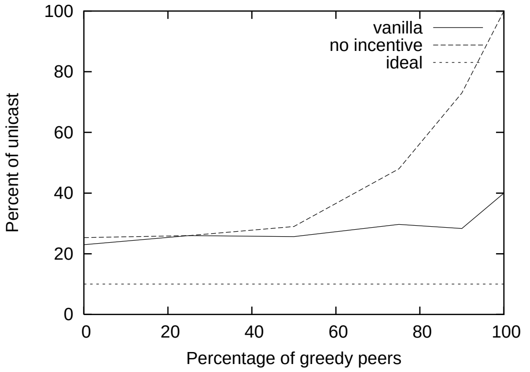

Figure 7: varying percent of greedy peers vs percent of unicast (10 hosts)

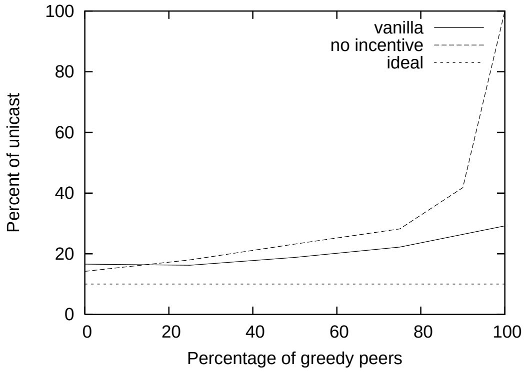

Figure 9: varying percent of greedy peers vs percent of unicast (20 hosts)

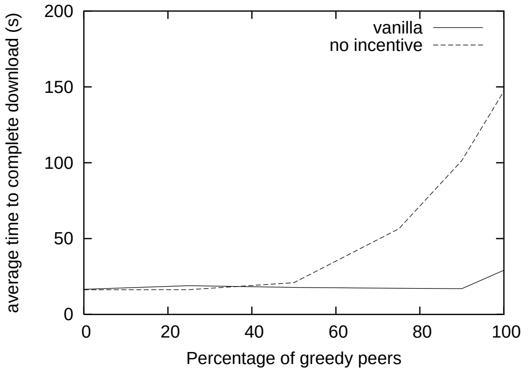

Figure 8: varying percent of greedy peers vs average download time (10 hosts)

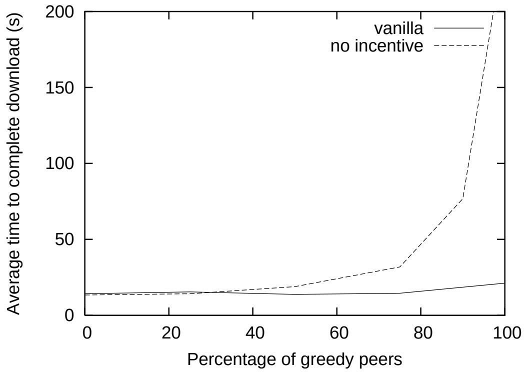

Figure 10: varying percent of greedy peers vs average download time (20 hosts)

no incentive BitTorrent.

This is probably because as the delay coefficient increases, it takes longer for all the peers to finish downloading the file. This means the seed is sending out data to peers for a longer period of time, so we would expect more bytes to be sent out and the percentage of unicast to be greater.

While this may be seen as a result implying that efficiency is "getting worse" as the delay coefficient increases, it is perhaps actually not the case. Since the average time to complete downloads decreases as the delay coefficient increases, what is happening can be seen as the network making better use of the resources of the seed as the percent of unicast increases.

Varying the number of hosts

In both vanilla BitTorrent and no incentive BitTorrent networks, performance improves with the number of hosts on the network, which is what we might have expected from a peer to peer network where (some) users are sharing resources.

Average download time

In most cases, as we vary the number of hosts in a trial, the average download time starts out high and decreases as the number of hosts increases. We are not sure why this happens, although it may be due to the fact that with fewer total hosts, there are also fewer cooperating (non-greedy) hosts on the network. Since the rate of non-greedy users in the network is constant (50%) and each additional non-greedy user can serve at least k = (minimum unchokes) other users, an increase in the number of cooperative hosts might more than make up for the same increase in the number of greedy hosts.

With only 2 hosts, we conjecture that the average download time is low because the hosts do not have much competition for the resources of the initial seed.

The average download times in the vanilla BitTorrent network were unaffected by the number of hosts on the network. As the number of hosts goes up, competition for the resources of the seed becomes fiercer, resulting in a decrease in expected gain from the source. However, this is offset by cooperation among trading peers who trade amongst themselves and avoid the bottleneck at the source, causing an increase in expected gain from the total of all the other peers on the network.

It seems that BitTorrent's incentive works to help the network as a whole avoid being held back by the bottleneck at the source and give lower average download times, particularly when the number of users is small.

Percentage of unicast

In both the vanilla BitTorrent and the no incentive BitTorrent, the percentage of unicast values decrease with the number of hosts. This is not too surprising since the best possible percentage of unicast (the ideal) decreases with the number of hosts as well. This means that the more hosts there are, the more they are trading amongst themselves and thus taking strain off the seed. The curves for both follow the best possible value of percentage of unicast pretty uniformly, and there's not much difference between them.

Varying the percentage of greedy hosts

Recall that the greedy strategy in vanilla BitTorrent is to upload to others while downloading and to leave the network (and not to continue uploading to others) as soon as the entire file is obtained. In no incentive BitTorrent, the greedy strategy is not to upload at all.

Vanilla BitTorrent outperformed no incentive BitTorrent in both measures of efficiency as the percentage of greedy hosts increased.

Average download time

Vanilla BitTorrent performed extremely well as the percentage of greedy hosts increased. The average time to finish downloading the file remained stable for almost any proportion of greedy peers. Even with all peers employing the greedy strategy, most of the observed average times to complete a download remained comparable to the times seen in trials of the vanilla BitTorrent with no greedy peers (a few outliers pull the average up significantly though).

With other default parameters, the average download times in the no incentive BitTorrent network remained comparable to the average download times in the vanilla BitTorrent network with up to about 50% of users utilizing the greedy strategy.

Percentage of unicast

The percentage of unicast sent out by the source also remained relatively stable (about 20% to 35%) in the vanilla BitTorrent network as the proportion of greedy peers increased. In the no incentive BitTorrent network, the percent of unicast sent out by the source remained comparable to the vanilla BitTorrent network for networks containing up to 50% of greedy peers as well.

Varying the size of the network and the percentage of greedy hosts

We also tried the suite of trials over different proportions of greedy peers on different sized networks. In the BitTorrent network, doubling the size of the network to 20 peers did not have much affect on the shape or height of the average download time curve as the proportion of greedy peers increased. The source had to send out relatively fewer bytes (as measured by percent of unicast), but the shape of the percentage of unicast curve was largely unaffected by doubling the size of the network as well.

Surprisingly, we observed that as the size of the network increased, the no incentive network was able to support a greater proportion of greedy peers. In the no incentive networks, both the average download time and percentage of unicast curves were observed to be more concave on a larger network. This means that a larger network can still perform reasonably well with a greater percentage of greedy peers, but once performance begins to suffer, things get bad really quickly as the percentage of greedy peers continues to increase.

6 Conclusions

The vanilla BitTorrent networks are indeed robust under the influence of greedy peers. The measurements of efficiency remained stable as we varied the proportion of greedy peers. In contrast, the no incentive networks performed comparably in the measures of efficiency up to a point, after which signs of strain quickly showed as we continued to increase the proportion of greedy peers. Surprisingly, the size of the network had an effect on the proportion of greedy peers that could be supported. The greater the number of peers on the network, the more greedy peers the network could support.

Vanilla BitTorrent networks turn out to be demonstrably more efficient than no incentive P2P networks. On average, the vanilla BitTorrent networks did at least as well as the no incentive networks under every setting, and it sometimes did much better. The BitTorrent incentives never hurt you, but you might be okay without incentives if the size of your network is large and the percentage of greedy peers is relatively small.

7 Remarks

The BitTorrent incentive has a limited scope. Because each BitTorrent network is for a single file, the incentive to upload to peers on one BitTorrent network will not carry over to another BitTorrent network where a different file can be traded. Since a user will likely connect to multiple networks during his lifetime, (by downloading more than one file), breaking down the incentive barriers between different BitTorrent networks would increase the efficacy of the incentive. We conjecture that the lack of a delayed reward is a major contributor to the shortfalls of BitTorrent: short content lifetime and lack of availibility of files [7].

One way that some BitTorrent trackers try to break down the barriers between file networks is to keep track of the upload/download ratio for each peer across all files. Peers with a low ratio may be penalized or prevented from using the tracker. It is then in a peer's best interest to continue to upload a file that it has finished downloading, because that will increase its ratio and allow it to download more in the future. However, until a global lasting form of delayed reward for good behavior is established, the BitTorrent incentive will continue to be limited in scope.

References

[1] Eytan Adar and Bernardo A. Huberman. Free riding on gnutella. *First Monday*, September 2000.

[2] Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. A game theoretic framework for incentives in p2p systems, 2003.

[3] Yang-Hua Chu, Sanjay G. Rao, and Hui Zhang. A case for end system multicast. In *Measurement and Modeling of Computer Systems*, pages 1–12, 2000.

[4] Bram Cohen. Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, June 2003.

[5] E. Elrufaie and D. Turner title =.

[6] Bennett Haselton. Algorithm for tracking credits in a peer-to-peer network. `http://www.peacefire.org/circumventor/peer-to-peer-credit-tracking-algorithm.html`, August 2004.

[7] Pouwelse, Barbacki, Epema, and Sips. A measurement study of the bittorrent peer-to-peer file-sharing system, May 2004. Preprint Submitted to Elsevier Science.

[8] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems, 2002.

[9] Jeffrey Shneidman, David C. Parkes, and Laurent Massoulie. Faithfulness in internet algorithms. In *Proc. SIGCOMM Workshop on Practice and Theory of Incentives and Game Theory in Networked Systems (PINS'04)*, Portland, USA, 2004.

[10] K. Tamilmani, V. Pai, and A. Mohr. Swift: A system with incentives for trading, 2004.

[11] V. Vishnumurthy, S. Chandrakumar, and E. Sirer. Karma: A secure economic framework for peer-to-peer resource sharing, 2003.