The Bittorrent P2P File-Sharing System: Measurements and Analysis

Johan Pouwelse, Paweł Garbacki, Dick Epema, and Henk Sips

Delft University of Technology, Delft, The Netherlands
j.a.pouwelse@ewi.tudelft.nl
Department of Computer Science, Parallel and Distributed systems group

Abstract. Of the many P2P file-sharing prototypes in existence, BitTorrent is one of the few that has managed to attract millions of users. BitTorrent relies on other (global) components for file search, employs a moderator system to ensure the integrity of file data, and uses a bartering technique for downloading in order to prevent users from freeriding. In this paper we present a measurement study of BitTorrent in which we focus on four issues, viz. availability, integrity, flashcrowd handling, and download performance. The purpose of this paper is to aid in the understanding of a real P2P system that apparently has the right mechanisms to attract a large user community, to provide measurement data that may be useful in modeling P2P systems, and to identify design issues in such systems.

1 Introduction

Even though many P2P file-sharing systems have been proposed and implemented, only very few have stood the test of intensive daily use by a very large user community. The BitTorrent file-sharing system is one of these systems. Measurements on Internet backbones indicate that BitTorrent has evolved into one of the most popular networks [8]. In fact, BitTorrent traffic made up 53 % of all P2P traffic in June 2004 [12]. As BitTorrent is only a file-download protocol, it relies on other (global) components, such as web sites, for finding files. The most popular web site for this purpose at the time we performed our measurements was suprnova.org.

There are different aspects that are important for the acceptance of a P2P system by a large user community. First, such a system should have a high availability. Secondly, users should (almost) always receive a good version of the content (no fake files) [10]. Thirdly, the system should be able to deal with flashcrowds. Finally, users should obtain a relatively high download speed.

In this paper we present a detailed measurement study of the combination of BitTorrent and Suprnova. This measurements study addresses all four aforementioned aspects. Our measurement data consist of detailed traces gathered over a period of 8 months (Jun'03 to Mar'04) of more than two thousand global components. In addition, for one of the most popular files we followed all 90,155 downloading peers from the injection of the file until its disappearance (several

M. Castro and R. van Renesse (Eds.): IPTPS 2005, LNCS 3640, pp. 205–216, 2005.

months). In a period of two weeks we measured the bandwidth of 54,845 peers downloading over a hundred newly injected files. This makes our measurement effort one of the largest ever conducted.

The contributions of this paper are the following: first, we add to the understanding of the operation of a P2P file-sharing system that apparently by its user-friendliness, the quality of the content it delivers, and its performance, has the right mechanisms to attract millions of users. Second, the results of this paper can aid in the (mathematical) modeling of P2P systems. For instance, in the fluid model in [13], it is assumed that the arrival process and the abort and departure processes of downloaders are Poisson, something that is in obvious contradiction with our measurements. One of our main conclusions is that within P2P systems a tension exists between availability, which is improved when there are no global components, and data integrity, which benefits from centralization.

2 The BitTorrent File-Sharing System

BitTorrent [5] in itself is only a file-downloading protocol. In BitTorrent, files are split up into chunks (on the order of a thousand per file), and the *downloaders* of a file *barter* for chunks of it by uploading and downloading them in a tit-for-tat-like manner to prevent parasitic behavior. Each peer is responsible for maximizing its own download rate by contacting suitable peers, and peers with high upload rates will with high probability also be able to download with high speeds. When a peer has finished downloading a file, it may become a *seed* by staying online for a while and sharing the file for free, i.e., without bartering.

Table 1. Popular BitTorrent web sites (Oct 2004)

Site name	Available files	File transfers
Suprnova.org	46,766	2,267,463
Youceff.com	47,137	1,145,889
Piratebay.org	39,294	749,133
Lokitorrent.com	30,957	816,435

To find a file in BitTorrent, users access web sites which act as global directories of available files. In Table 1, we show for the most popular of these web sites the number of different files and the number of active file transfers at a certain time. In december 2004, the top two of these web sites went offline. In this paper we assume Suprnova as the directory web site.

The Suprnova web site uses a *mirroring system* to balance user requests across its mirror sites. The web pages on Suprnova show for each available file the name and size, the current numbers of downloaders and seeds, and the name of the person who uploaded the file. To start the download of a file, a user clicks on a link pointing to `.torrent` meta-data file. These meta-data files are not stored on Suprnova or its mirrors, but are distributed among a number of `.torrent` file servers. In turn, each `.torrent` file points to a *tracker*, which keeps a global registry of all the downloaders and seeds of the corresponding file. The tracker responds to a user's request with a list of some of the peers having (part of) the requested file, with whom the user can establish direct connections to barter for chunks of the file. One tracker can supervise the simultaneous downloads of multiple files.

New content is injected into BitTorrent by uploading a `.torrent` file to the Suprnova web site and creating a seed with the first copy of the file. In order to reduce the pollution level, new content is first manually inspected by *moderators*, who weed out fake content, content with low perceptual quality, and content with incorrect naming. A normal user who injects content is called a *moderated submitter*. To lower the burden on the moderators, a user who frequently injects correct content is promoted to the rank of *unmoderated submitter*, and is allowed to directly add content. Unmoderated submitters can request a promotion to moderator status to existing moderators.

Together, BitTorrent and Suprnova form a unique infrastructure that uses mirroring of the web servers with its directory structure, meta-data distribution for load balancing, a bartering technique for fair resource sharing, and a P2P moderation system to filter fake files.

3 Experimental Setup

In this section, we will discuss some details of our measurement software and the collected data. Our measurement software consists of two parts with three scripts each. The first part is used for monitoring the global BitTorrent/Suprnova components, and consists of the *Mirror script* which measures the availability and response time of the Suprnova mirrors, the *HTML script* which gathers and parses the HTML pages of the Suprnova mirrors and downloads all new `.torrent` files, and the *Tracker script* which parses the `.torrent` files for new trackers and checks the status of all trackers.

The second part of our software is used for monitoring actual peers. To follow thousands of peers at one minute time resolution we used 100 nodes of our Distributed ASCI Supercomputer (DAS, `cs.vu.nl/das2`). The *Hunt script* selects a file to follow and initiates a measurement of all the peers downloading this particular file, the *Getpeer script* contacts the tracker for a given file and gathers the IP addresses of peers downloading the file, and the *Peerping script* contacts numerous peers in parallel and (ab)uses the BitTorrent protocol to measure their download progress and uptime. The Hunt script monitors once per minute every active Suprnova mirror for the release of new files. Once a file is selected for measurement, the Getpeer and Peerping scripts are also activated at the same time resolution. In this way we are able to obtain the IP addresses of the peers that inject new content and we can get a good estimate of the average download speed of individual peers.

In doing our measurements, we experienced three problems. First, our measurements were hindered by the wide-spread usage of firewalls [11]. When a peer is behind a firewall, our Getpeer script can obtain its IP number, but the Peerping script cannot send any message to it. Therefore, our results for download speed are only valid for non-firewalled peers. The second problem was our inability to obtain all peer IP numbers from a tracker directly. The BitTorrent protocol specifies that a tracker returns only a limited number (with a default of 20) of randomly selected peer IP numbers. We define the *peer coverage* as the fraction of all peers that we actually discovered. In all our measurements we obtained a peer coverage of over 95 %. Our final measurement problem was caused by modifications made to the BitTorrent system itself. Which created minor gaps in our traces.

4 Measurement Results

In this section, we first show the number of users downloading or seeding on BitTorrent/Suprnova. Then we present detailed performance measurements of the availability, the integrity, the flashcrowd effect, and the download performance of the system.

4.1 Overall System Activity

The number of users over time on BitTorrent/Suprnova gives a good indication of both the general performance and the dynamics of the system. We show the popularity of BitTorrent/Suprnova in terms of the number of downloads over time and its dependence on technical failures in the system.

Figure 1 shows the total number of downloads, and the number of downloads of three types of content (games, movies, and music) in progress in BitTorrent around Christmas 2003. We have selected this month for presentation because it shows

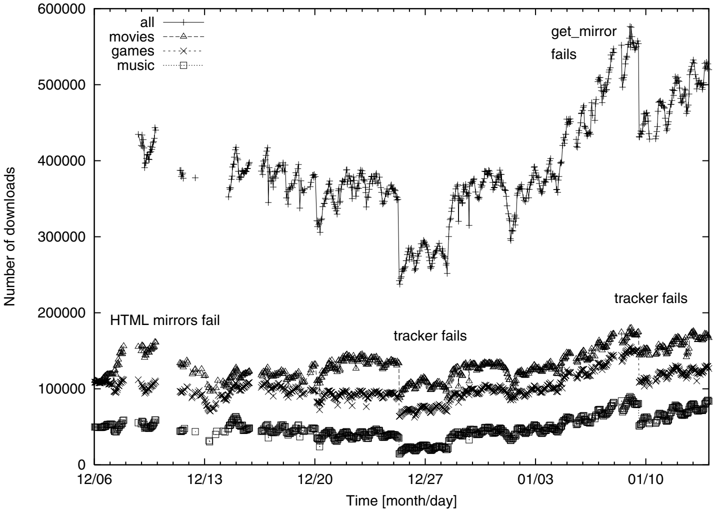

Fig. 1. The number of users downloading or seeding on BitTorrent/Suprnova for one month (Dec'03-Jan'04)

a large variance in the number of downloads due to several BitTorrent/Suprnova failures. The lowest and highest number of downloads in Figure 1 are 237,500 (on Christmas day) and 576,500 (on January 9). Our HTML script requests every hour all pages from one of the active Suprnova mirrors. The consecutive data points have been connected with a line when there was no overall systems failure.

There are two things to be noted in Figure 1. The first is the daily cycle; the minimum and maximum (at around 23:00 GMT) number of downloads occur at roughly the same time each day, which is similar to the results found in [14]. The second is the large variation due to failures of either the mirroring system across the Suprnova mirrors, of the mirrors themselves, of the `.torrent` servers, or of the trackers. For example, on December 8 and 10, gaps occurred due to failures of the mirroring system and of 6 out of 8 Suprnova mirrors, and on Christmas day, a large tracker went off-line for 98 hours. The failure of this single tracker alone reduced the number of available movies from 1675 to 1017, and resulted in a sharp reduction in the number of downloads. From January 5 to 10, the mirroring system was also off-line a few times, causing `suprnova.org` to be unusable and the Suprnova mirrors not being updated, which is visible in the figure as a few gaps in the "all" line. The figure suggests that users are not discouraged by such failures.

We conclude that the number of active users in the system is strongly influenced by the availability of the global components in BitTorrent/Suprnova.

4.2 Availability

In this section we present measurements of the availability of both the global Suprnova components and the BitTorrent peers.

The BitTorrent/Suprnova architecture is vulnerable because of potential failures of the four types of *global components.* The main `suprnova.org` server sometimes switched IP number and was down several times. The various mirrors rarely survive longer than a few days due to the high demands of over 1,200,000 daily visitors (Oct 2004), and sometimes, fewer than five mirrors were up. Occasionally, no `.torrent` file servers were available, blocking all new downloads. In general, trackers are a frequent target for denial-of-service attacks and are costly to operate due to GBytes of daily bandwidth consumption.

Figure 2 shows the results of our availability measurements of 234 Suprnova mirrors, 95 `.torrent` file servers, and 1,941 BitTorrent trackers (Suprnova.org itself is not shown). In the figure we plot the average uptime in days for these global components ranked according to decreasing uptime. Only half of the Suprnova mirrors had an average uptime of over 2.1 days, which is a good indication of their (un)availability. In addition, only 39 mirrors had a continuous uptime period longer than two weeks. We can conclude that reliable webhosting of Suprnova pages is a problem. As shown in the figure, the `.torrent` file servers are even less reliable. A few trackers show a high degree of availability, with one tracker even showing a continuous uptime period of over 100 days. Half of the trackers had an average uptime of 1.5 day or more, and the 100 top ranking trackers had an average uptime of more than 15.7 days.

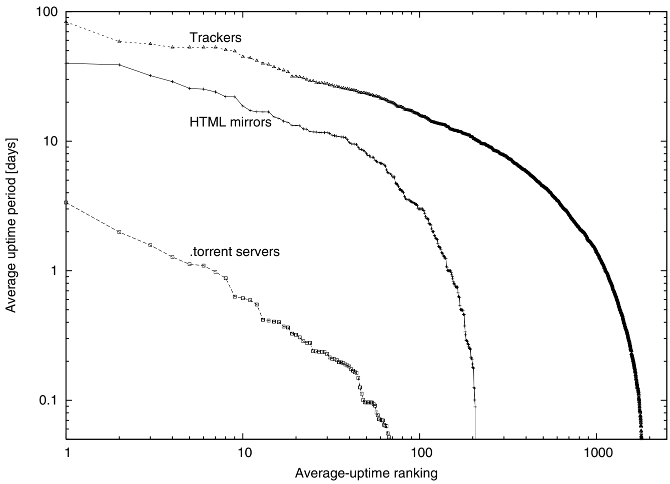

Fig. 2. The uptime ranking of three types of BitTorrent/Suprnova global components

In Figure 1 we have shown that unavailability has a significant influence on popularity. Combined with the high frequency of such failures as apparent from Figure 2, we conclude that there is an obvious need to decentralize the global components. However, all the features that make BitTorrent/Suprnova exceptional (easy single-click-download web interface, low level of pollution, and high download performance) are heavily dependent on these global components.

The availability of *individual peers* over a long time period has never been studied, despite its importance. We measured peer availability for over three months, which is significantly longer than reported in [2], [4], and [14].

On December 10, 2003 the popular PC game "Beyond Good and Evil" from Ubisoft was injected into BitTorrent/Suprnova and on March 11, 2004 it died. We followed this content and obtained 90,155 peer IP numbers using our Getpeer script. Of these IP numbers, only 53,883 were not behind firewalls and could be traced by our Peerping script. We measured the uptime of all non-firewalled peers with a one minute resolution.

Figure 3 shows the results of our uptime measurements. Here we plot the peer uptime in hours *after* they have finished downloading with the peers ranked according to decreasing uptime. The longest uptime is 83.5 days. Note that this log-log plot shows an almost straight line between peer 10 and peer 5,000. The sharp drop after 5,000 indicates that the majority of users disconnect from the system within a few hours after the download has been finished. This sharp drop has important implications because the actual download time of this game spans several days. Figure 3 shows that seeds with a high availability are rare. Only 9,219 out of 53,883 peers (17 %) have an uptime longer than one hour after they finished downloading. For 10 hours this number has decreased to only 1,649 peers (3.1 %), and for 100 hours to a mere 183 peers (0.34 %).

Our two availability figures depict crucial information for architectural improvements. To increase the availability of the whole system, the functionality

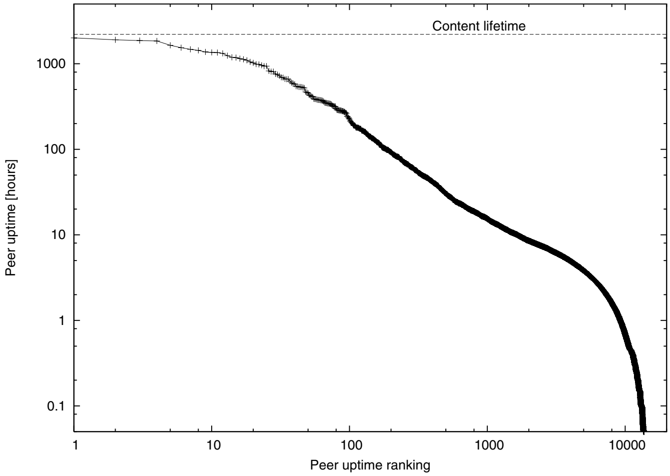

Fig. 3. The uptime distribution of the 53,833 peers downloading "Beyond Good and Evil"

of the global components would have to be distributed, possibly across the ordinary peers. However, as peers with a high uptime are currently very rare, then peers should be given incentives to lengthen their uptimes.

4.3 Integrity

This section analyses the integrity in BitTorrent/Suprnova of both the content itself and of the associated meta-data, which is a notorious problem in P2P systems.

In order to test the integrity of meta-data, we donated to Suprnova an account for hosting a mirror. By installing spyware in the HTML code, we have registered each `.torrent` download and could have easily corrupt the meta-data. We conclude that using donated resources for hosting meta-data entails substantial integrity and privacy risks.

As to the integrity of the content, P2P message boards and other sources strongly indicate that BitTorrent/Suprnova is virtually pollution free. However, a direct measurement of fake or corrupted files is difficult; manually checking the content of many files is not really a viable option. Instead, we actively tried to pollute the system. We created several accounts on different computers from which we tried to insert files that were obviously fake. We failed; the moderators filtered out our fake files.

The system of moderators seems to be very effective in removing fake and corrupted files. The following measurements show that only a few of such volunteers are needed. Figure 4 shows the numbers of files that have been injected by the 20 moderators, the 71 unmoderated submitters, and the 7,933 moderated submitters that were active between June 2003 and March 2004. The ten most active moderated submitters injected 5,191 files, versus 1,693 for the unmoderated submitters and 274 for the moderators. We are surprised that a

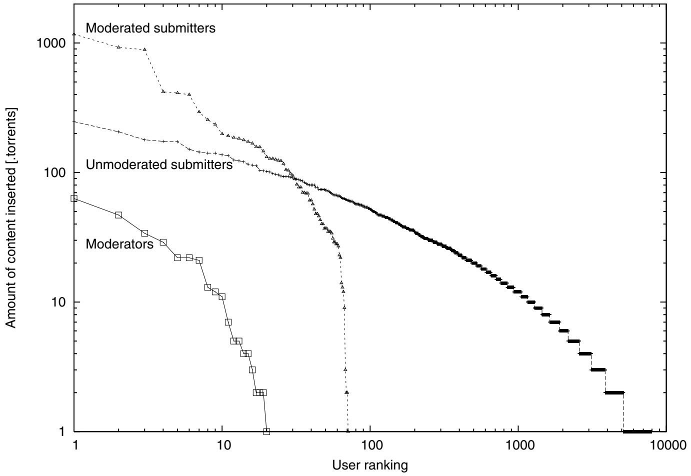

Fig. 4. The activity of the different content submitters on Suprnova to prevent pollution

mere 20 moderators were able to effectively manage the numerous daily content injections with such a simple system. Unfortunately, this system of moderation relies on global components and is extremely difficult to distribute.

4.4 Flashcrowds

We now focus on the system's reaction to the sudden popularity of a single (new) file. This phenomenon is also called the *flashcrowd* effect. Figure 5 shows the number of downloads for a single file as a function of time (the Lord of the Rings III movie with size 1.87 GByte). We have selected this file because it uses a tracker (FutureZone.TV) which provides access to detailed statistics, which we collected every five minutes with our Tracker script. The top line shows the sum of the number of downloads in progress and the number of seeds according to this tracker, while the bottom line only shows the number of seeds. During the first five days, no peer finished downloading the file and the injector of the file was continuously online. This long time period provides a clear opportunity to identify copyright violators. The statistics from Suprnova were fetched by our HTML script every hour, and are in agreement with the total tracker results to such an extent that the lines overlap almost completely. Only on December 23, 2003 there was a problem with the tracker for a few minutes, which is not visible in the Suprnova data. The results from the Peerping script show a significantly lower number of downloads, which is due to the firewall problem (40 % of the peers were firewalled). The gaps in the Peerping results were due to disk quota problems on the DAS, which ran our measurement software. From the measurements we conclude that the global BitTorrent/Suprnova components are capable of efficiently handling very large flashcrowds. Also, because of the strong sudden increase in the number of downloaders, it is clear that the arrival process is not Poisson.

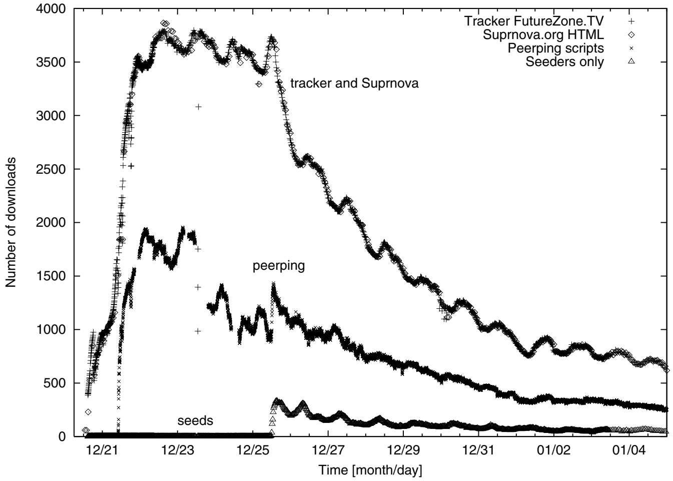

Fig. 5. Flashcrowd effect of "Lord of the Rings III"

4.5 Download Performance

In this section, we examine the efficiency (download speed) and the effectiveness (number of available files) of downloading.

Figure 6 presents the results of a two-week experiment in which the average download bandwidth of 54,845 peers was measured. To obtain these measurements, our Hunt script followed the first 108 files that where added to Suprnova on March 10, 2004. The figure also shows the Cumulative Distribution Function (CDF) of the fraction of peers with a certain download speed. It turns out that 90% of the peers had a download speed below 520 kbps; the average download

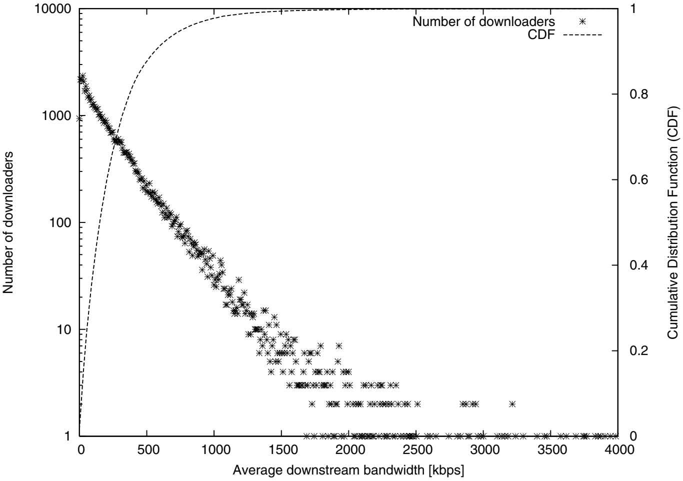

Fig. 6. The average download speed of peers

speed of 240 kbps allowed peers to fetch even large files in one day. An important observation is the exponential relation between the average download speed and the number of downloads at that speed.

In BitTorrent the availability of content is unpredictable. When the popularity drops and the last peer/seed with certain content goes offline, the content dies. Figure 7 shows the content lifetime of all large files (at least 500 MByte) on BitTorrent/Suprnova we have followed. Each file is represented as a data point with on the horizontal axis the number of seeds for the file 10 days after its injection time, and on the vertical axis its content lifetime. Important observations are that the number of seeds after 10 days is not an accurate predictor for the content lifetime, and that files with only a single seed can still have a relatively long content lifetime.

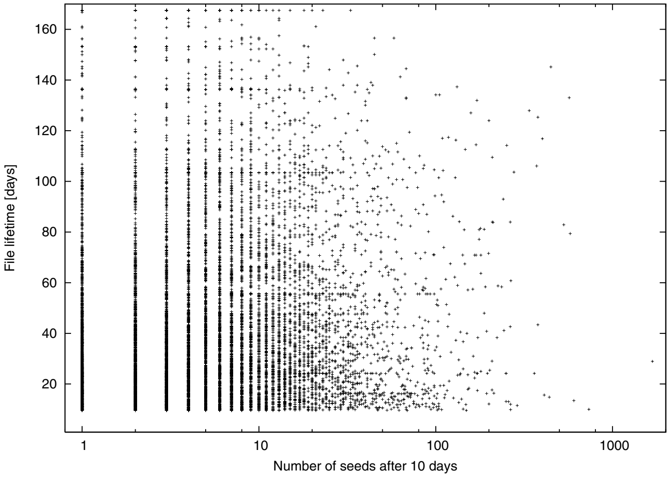

Fig. 7. The content lifetime versus the number of seeds after 10 days for over 14,000 files

BitTorrent itself does not have incentives to seed, but there are Suprnova-like websites that do so. For instance, the software from `Yabtuc.org` consists of such a website with an integrated tracker which registers seeding behaviour. When users do not upload sufficiently, their access is temporarily denied. However, this system is even more centralized then Suprnova due to the integration of website and tracker at a single location.

5 Related Work

Previous work on BitTorrent has focused on measurements [5,12,8,7], theoretical analysis [13], and improvements [16]. In [7], the log of a Bittorrent tracker is analysed; it shows for a single file the flashcrowd effect and download speed. In [13] a fluid model is used to determine the average download time of a single file. This remarkable model assumes Poisson arrival and departure processes for downloaders and seeds, equal upload and download bandwidths for all peers, and no flashcrowd effect. However, their assumption of Poisson processes is contradicted by the results of this paper, indicating the strong need for proper workload characterization to validate P2P models.

Improvements to BitTorrent-like software are presented in [16]. Their system effectively decentralizes the tracker. However, due to the complete lack of integrity measures it will be trivial to corrupt this system.

For other P2P systems than BitTorrent, several measurement studies of P2P networks have addressed the issues of availability [2,4,6], integrity [10], flashcrowds [5,9], and download performance [1,15,14,3,4]. Most of the availability studies only span a few days [2] or weeks [4], making it difficult to draw conclusions on long-term peer behavior. The only long-term study is a 200-day trace of the Kazaa traffic on the University of Washington backbone [6], but the well-connected users with free Internet access in this environment are not average P2P users. Integrity of P2P systems has received little attention from academia. A unique study found that for popular songs on Kazaa, up to 70 % of the different versions are polluted or simply fake [10]. The Kazaa moderation system based on voting is therefore completely ineffective. In one of the first studies (August 2000) related to download performance [1], over 35,000 Gnutella peers where followed for one day. Nearly 70 % of the peers did not contribute *any* bandwidth. In [15] it is found that less than 10 % of the IP numbers fill about 99 % of all P2P bandwidth. In [14], SProbe (`sprobe.cs.washington.edu`) was used to measure the bandwidth of 223,000 Gnutella peers in May 2001. It turned out that roughly 8 % of the Gnutella peers downloaded files at speeds lower than 64 kbps.

Content lifetime is still a poorly understood and unexplored research area. Only one paper has investigated when content appeared on a P2P network, but not when it disappeared [3].

6 Discussion and Conclusions

In this paper we have presented a detailed measurement study and an analysis of the BitTorrent/Suprnova P2P system. We believe that this study is a contribution to the ongoing effort to gain insight into the behavior of widely used P2P systems. In order to share our findings we have published all raw data files (anonymized), measurement software, and documentation on `peer-2-peer.org`.

One of the big advantages of BitTorrent/Suprnova is the high level of integrity of both the content and the meta-data due to the working of its global components. We have shown that only 20 moderators combined with numerous other volunteers solve the fake-file problem on BitTorrent/Suprnova. However, this comes at a price: system availability is hampered by the global nature of these components. Decentralization would provide an obvious solution, but makes the meta-data more vulnerable. Also, a decentralized scheme such as in Kazaa has no availability problems but lacks integrity, since Kazaa is plagued with many fake files. Clearly, decentralization is an unsolved issue that needs further research.

Another future design challenge for P2P file sharing is creating incentives to seed. For example, peers that seed files should be given preference to barter for other files.

References

1. E. Adar and B. A. Huberman. Free riding on gnutella. Technical report, Xerox PARC, August 2000.
2. R. Bhagwan, S. Savage, and G. M. Voelker. Understanding availability. In *International Workshop on Peer to Peer Systems*, Berkeley, CA, USA, February 2003.
3. S. Byers, L. Cranor, E. Cronin, D. Kormann, and P. McDaniel. Analysis of security vulnerabilities in the movie production and distribution process. In *The 2003 ACM Workshop on DRM*, Washington, DC, USA, Oct 2003.
4. J. Chu, K. Labonte, and B. Levine. Availability and locality measurements of peer-to-peer file systems. In *ITCom: Scalability and Traffic Control in IP Networks*, Boston, MA, USA, July 2002.
5. B. Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, USA, May 2003. `http://bittorrent.com/`.
6. K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *19-th ACM Symposium on Operating Systems Principles*, Bolton Landing, NY, USA, October 2003.
7. M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. Al Hamra, and L. Garces-Erice. Dissecting bittorrent: Five months in a torrent's lifetime. In *Passive and Active Measurements*, Antibes Juan-les-Pins, France, April 2004.
8. T. Karagiannis, A. Broido, N. Brownlee, kc claffy, and M. Faloutsos. Is p2p dying or just hiding? In *Globecom*, Dallas, TX, USA, November 2004.
9. N. Leibowitz, M. Ripeanu, and A. Wierzbicki. Deconstructing the kazaa network. In *3rd IEEE Workshop on Internet Applications (WIAPP'03)*, San Jose, CA, USA, June 2003.
10. J. Liang, R. Kumar, Y. Xi, and K. Ross. Pollution in p2p file sharing systems. In *IEEE Infocom*, Miami, FL, USA, March 2005.
11. T. Oh-ishi, K. Sakai, T. Iwata, and A. Kurokawa. The deployment of cache servers in p2p networks for improved performance in content-delivery. In *Third International Conference on Peer-to-Peer Computing (P2P'03)*, Linkoping, Sweden, September 2003.
12. A. Parker. The true picture of peer-to-peer filesharing, 2004. `http://www.cachelogic.com/`.
13. D. Qiu and R. Srikant. Modeling and performance analysis of bit torrent-like peer-to-peer networks. In *ACM SIGCOMM*, Portland, OR, USA, August 2004.
14. S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Multimedia Computing and Networking (MMCN'02)*, San Jose, CA, USA, January 2002.
15. S. Sen and J. Wang. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Transactions on Networking*, 12(2):219–232, 2004.
16. R. Sherwood, R. Braud, and B. Bhattacharjee. Slurpie: A cooperative bulk data transfer protocol. In *IEEE Infocom*, Honk Kong, China, March 2004.