A Measurement Study of the BitTorrent Peer-to-Peer File-Sharing System

J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips

Parallel and Distributed Systems group, Delft University of Technology, The Netherlands
P.O. box 5031, 2600GA Delft, ITS, room 09.250, Tel +31 152782539, Fax +31 152786213

Abstract

P2P systems for sharing content have become very popular over the last few years. However, despite the increasing attention of both the research community and large numbers of users, the actual behavior of these systems over prolonged periods of time is still poorly understood. This paper presents a detailed measurement study over a period of eight months of BitTorrent/Suprnova, a P2P file-sharing system that is quickly gaining in popularity. In particular, we show measurement results of the popularity and the availability of BitTorrent, of its download performance, of the content lifetime, and of the structure of the community responsible for verifying uploaded content. We also propose improvements to BitTorrent to increase its availability and performance.

Key words: peer-to-peer, file sharing, BitTorrent, performance measurements

1 Introduction

Over the last few years, quite a few peer-to-peer (P2P) protocols for sharing files (e.g. containing music or movies) have been devised. However, despite the increasing attention from both the research community and hundreds of thousands of users,

Email address: `j.a.pouwelse@its.tudelft.nl` (J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips).

the behavior of such systems under a real workload of actual users is poorly understood. One of the most promising current P2P file-sharing systems that is quickly gaining popularity, is the BitTorrent/Suprnova system [1]. This paper is an attempt to understand the operation of P2P file sharing by means of detailed measurements of the use and operation of BitTorrent to assess its quality with respect to such aspects as download speed and availability. In addition, we propose improvements to the existing architecture.

A high level of robustness and the "one-click download" user interface are important factors in the success of BitTorrent. BitTorrent is different from other P2P networks in three important respects. First, it does not include a search mechanism, but rather, it relies on central-directory based search facilities as provided by Web sites such as `suprnova.org`, which maintain lists of all files currently available for downloading. Secondly, it employs a file-level sharing policy instead of the common directory-level sharing policy. Thirdly, it provides a bartering mechanism among the clients who are downloading the same file, which introduces a certain level of fairness into the system.

In this paper we present a detailed measurement study of BitTorrent in order to get more insight into the technical and collective user behavior of the system. We do this by looking at a number of common aspects that characterize the behavior of file-sharing P2P systems, viz. the popularity, the availability, the download speed, the content lifetime, and the community responsible for verifying uploaded content in BitTorrent. Our measurement data consist of detailed traces gathered over a period of eight months. Our focus in this paper is on the usefulness of P2P concepts for the rapid dissemination of information among large numbers of computers—the fact that also BitTorrent contains large amounts of illegally obtained and shared content is not relevant in this context.

Furthermore, we make a comparison of BitTorrent with four other P2P architectures, namely FastTrack (`Kazaa.com`), Overnet (`overnet.com`), Direct Connect (`neo-modus.com`), and Gnutella (`limewire.com`). Finally, from our measurements of BitTorrent we identify weak points for which we propose improvements.

2 BitTorrent and Suprnova

In this section we provide an overview of the BitTorrent/Suprnova P2P system [1]. The architecture of the system is depicted in Figure 1.

BitTorrent does not provide a file-search mechanism itself; instead, users have to go to web sites which act as central directories listing recently released files. Suprnova (`suprnova.org`), with around 800,000 visitors per day, is currently the dominating web site for this purpose.

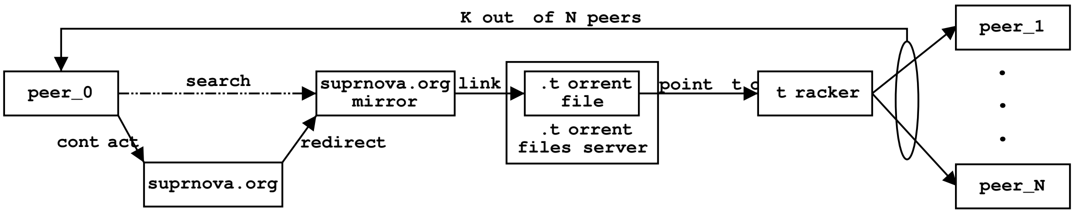
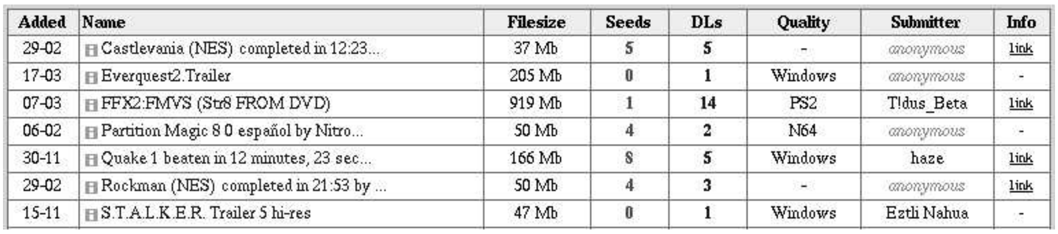


Fig. 1. The BitTorrent/Suprnova architecture.

To start the download of a file, a user (`peer_0`) first has to obtain the corresponding `.torrent` meta-data file. In order to do so, the user contacts the web site `suprnova.org`, whose sole purpose it is to balance the user requests across the Suprnova mirror sites (we will refer to this mechanism as the *mirroring system*). These sites present the user with a list such as shown in Figure 2 of (pointers to) `.torrent` files (here representing games). The meaning of the columns is, from left to the right: the date of upload, the name of the file, the file size, the number of seeds (peers that possess the whole file), the number of peers that are registered at the tracker (see below), the operating system for which the game is designed, the name of the person who uploaded the `.torrent` file, and the link to the web page with a description of the game. The meta-data files are not stored on Suprnova mirrors themselves, but are distributed among a number of file servers. The user then clicks on a link pointing to a location of the `.torrent` file.



Added	Name	Filesize	Seeds	DLs	Quality	Submitter	Info
29-02	Castlevania (NES) completed in 12:23...	37 Mb	5	5	-	anonymous	link
17-03	Everquest2.Trailer	205 Mb	0	1	Windows	anonymous	-
07-03	FFX2:FMVS (Str8 FROM DVD)	919 Mb	1	14	PS2	T!dus_Beta	link
06-02	Partition Magic 8 0 español by Nitro...	50 Mb	4	2	N64	anonymous	-
30-11	Quake 1 beaten in 12 minutes, 23 sec...	166 Mb	8	5	Windows	haze	link
29-02	Rockman (NES) completed in 21:53 by ...	50 Mb	4	3	-	anonymous	link
15-11	S.T.A.L.K.E.R. Trailer 5 hi-res	47 Mb	0	1	Windows	Eztli Nahua	-

Fig. 2. Several lines from a `suprnova.org` web page.

The `.torrent` file indicated by the user points to a *tracker*, which, as its name suggests, keeps track of all the peers which are (or recently were) also in the process of downloading the corresponding content, and which responds to the user's request with a list of (some of) these peers. From this point onwards, the user can establish direct connections with these peers participating in the download and start bartering chunks of the requested file with them. One tracker can supervise the simultaneous downloads of multiple files, and different `.torrent` files corresponding to the same data file may link to different trackers. In addition to the location of a tracker, `.torrent` files also contain content hashes that may be used to verify the correctness of the incoming data. It should be noted that even though there may exist many of each of the three types of central components (mirrors, file servers, and trackers), they are not replicas that take over each others' role when one of them goes down.

The key philosophy of BitTorrent is that peers should barter for chunks of files, i.e.

upload content at the same time they are downloading it. This bartering diminishes parasitic behavior of users—it is not possible to download without sharing. Each peer is responsible for maximizing its own download rate. Peers do this by downloading from whoever they can and deciding on which peers to upload to, via a variant of tit-for-tat in which a peer responds in one period with the same action that its collaborator performed in the previous period. As a result, peers with high upload rates will probably also be able to download with a high speed. In this manner BitTorrent can achieve a high bandwidth utilization. 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.

New content is injected into BitTorrent by uploading a `.torrent` file to `suprnova.org` and creating a seed with the first copy of the file. However, 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 add content to BitTorrent. Unmoderated submitters can request a promotion to moderator status. We call this a P2P moderation system because as is based on volunteers, and potentially everybody can become a moderator. As shown in Figure 2, the person who injected a file is clearly visible.

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 Approach and related work

In this section, we first define the five characteristics of P2P systems for which we will present measurements of BitTorrent in Section 5. Then we review related work in the form of a comparison of BitTorrent with four other popular P2P file-sharing systems with respect to these characteristics, and we discuss other P2P measurement studies.

3.1 *Characteristics of* P2P *systems*

We base our treatment of BitTorrent on the following five characteristics:

(1) The *popularity* of a P2P system is understood as the total number of users participating in it over a certain period of time.

(2) The *availability* is of paramount importance in P2P systems. As these systems have no (or sometimes only a few, such as in BitTorrent) central components, the operation generally depends on voluntarily contributed resources, which easily leads to low system availability.

(3) The *download performance* determines the relation between the size of a file and the time needed for downloading it.

(4) Several aspects of the content provided by a particular P2P network are very important to its users, such as the amount and types of content it provides. However, in this paper we focus in particular on the *content lifetime*, which is the time period that a file can actually be retrieved from the system, that is, the period from when it is injected into the system until none of the peers in the system is willing to share the file anymore. Another dynamic aspect of content is the *content injection time*, which is the time period between the first

official release of the content anywhere in the world (e.g. a movie release) and the moment it is injected into a P2P system.

(5) The *pollution level* of a P2P system is defined as the fraction of corrupted or wrong content. All files on P2P networks are voluntarily provided, and there is usually no authentication or verification of their contents. Therefore, anybody can, either intentionally or unintentionally, upload files with incorrect names or corrupted contents (e.g. fake copies of songs), which may subsequently spread throughout the system.

3.2 A comparison of popular P2P *file-sharing systems*

In this section we provide a comparison of BitTorrent with four of the most successful P2P systems with respect to the five characteristics presented in Section 3.1. These systems are FastTrack, which is the basis of Kazaa, Overnet (including eDonkey), DirectConnect (DC), and Gnutella. Table 1 shows the strong and weak points of these P2P systems, which are based on measurement surveys [3,4,6], file-sharing portals (e.g. `slyck.com`), and our own experiences with these systems. An important conclusion to be drawn from this table is that no single characteristic explains the popularity of the P2P systems.

P2P system	Strong points	Weak points
FastTrack	popularity, availability, content lifetime	pollution level
Overnet	popularity, content lifetime	download performance
BT/Suprnova	popularity, download performance, content injection time, pollution level	availability, content lifetime
DC	download performance, content lifetime	availability
Gnutella	download performance	popularity, pollution level

Table 1
A comparison of the five most popular P2P systems by means of five characteristics.

Concerning the popularity of the systems, FastTrack has the largest file sharing community, although Overnet does not stay far behind and BitTorrent is rapidly gaining popularity. DC was the first system to fully exploit social relationships by using online friendships and communities. In DC, users have to connect to a hub to be able to share and download files. The concept of search hubs and minimum file sharing requirements on individual hubs in combination with a cumbersome interface is reflected in a low popularity of this system. The popularity of Gnutella falls as quickly as it grew a few years ago; its original flat-flooding design does not scale well.

FastTrack is an architecturally advanced P2P system, and has good availability due to its utilization of supernodes to create temporary indexing servers, allowing the network to scale very well. Other systems try to ensure availability by means of full (Gnutella) or partial (Overnet) distribution of the responsibility for the shared files. The manually maintained Suprnova site does not scale by definition and is therefore one of the weakest points of BitTorrent.

BitTorrent is the indisputable leader in download performance. Its lack of searching functionality is compensated by an advanced download distribution protocol that leaves the competitors far behind. Overnet takes a completely opposite approach by offering powerful searching capabilities and queue-based scheduling of downloads with waiting times sometimes exceeding a few days.

The big difference between BitTorrent and other systems presented in Table 1 is the file-sharing policy. BitTorrent provides a file-level sharing scope by allowing users to download files only if these are already being downloaded. The lack of archive functionality results in relatively short content lifetimes. Systems like Kazaa use a directory-level sharing policy in which all the files located in a particular directory are accessible as long as the client stays connected. Hence, a temporary drop of file popularity does not remove it from the system.

Filtering of the shared content in order to create a pollution-free system is a process that can hardly be automated. Kazaa users are familiar with problems caused by fake files floating around in the system. Hash-code verification in combination with hash databases like `verifieds.com` do not limit the number of fake files in the FastTrack network, they allow user to identify correct files. The operation of Suprnova differs in this respect, as it prevents fake files from entering the system entirely.

3.3 Other measurement studies

Several studies have measured the five characteristics of P2P networks mentioned in Section 3.1, but most of these only span a few days, making it difficult to draw conclusions on long-term peer behaviour. The only long-term study is a 200-day trace of the Kazaa traffic on the University of Washington backbone [7]. However, the well-connected users with free Internet access in this environment are not average P2P users.

In [11], popularity measurements of Kazaa, DC, and Gnutella are described, and a detailed analysis of P2P traffic on the AT&T backbone is presented. The number of unique IP numbers involved in Kazaa traffic increased from 3,403,900 in September 2001 to 5,924,072 in December 2001, indicating a significant rise of Kazaa's popularity.

The availability of 2,400 Overnet clients during a 7-day period in January 2003 is studied in detail in [5]. A problem with such studies is that peers may change IP number between sessions, which was overcome by exploiting a unique identifier in the Overnet P2P client. It was found that 32 % of the responding peers used five or more IP numbers during the measurement period. In [10] the availability of a few thousand Gnutella peers is measured during a six-week period. It was observed that about 31 % of the uptimes of the peers had a duration of roughly 10 minutes (their measurement resolution).

In one of the first studies (August 2000) related to download performance [2], over 35,000 Gnutella peers where followed for one day. Nearly 70 % of the peers did not contribute *any* bandwidth, which is called "freeriding". Freeriding is also found in [11], namely less than 10 % of the IP numbers fill about 99 % of all P2P bandwidth. In [6], 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 a speed lower than 8 KB/s.

Content lifetime and content injection time are a poorly understood and unexplored research area. In [8], which is the only study of content injection time, an unnamed web site similar to `verifieds.com` is used to download 183 of the most popular Hollywood movies from an unnamed P2P system (not BitTorrent/Suprnova). It was discovered that 77 % of these movies were leaked by movie industry insiders and that most movies appeared on the network around their cinema or DVD release date. The most popular files on the Gnutella network are measured in [10] and the top-50 files are listed. An initial measurement of the flash crowd effect (a sudden increase in interest) is presented in [1]. This flash crowd was caused by a few thousand volunteers who were asked by the author of BitTorrent to download a test file. In [4], the variation in popularity of files over their lifetime in Kazaa is analyzed. Flash crowds are not found on Kazaa, probably because it does not include features to display the latest content such as included in Suprnova.

Pollution on P2P systems has also received little attention from academia. We believe that our measurements are the first attempt to accurately quantify some of its properties. The American company OverPeer has patented their method to damage the sound quality of audio files on P2P systems. In [12], the director of Overpeer states that they are actively polluting P2P networks with fake (MP3) content to protect over 30,000 titles.

4 Experimental setup

We have performed measurements of BitTorrent for all five characteristics discussed in Section 3.1. In this section we will discuss some details of our measurement software and the collected data. We have collected statistics from BitTorrent/Suprnova from June 2003 onwards and continue to do so. This article uses the statistics up to March 2004, which amount to over 50 GBytes of raw data. We used 100 nodes of our Distributed ASCI Supercomputer (DAS, `cs.vu.nl/das2`) to measure every minute the download bandwidth of more than 100,000 downloads of popular content such as the ”Matrix Revolutions”, ”Lord of the Rings III”, and ”Beyond Good and Evil”, and to obtain the statistics of over 60,000 files (first and last appearance, number of downloads with a resolution of one hour).

Our measurement software consists of two parts with three scripts each. The first part is used for monitoring the central 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 the actual peers in the system. 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 every active Suprnova mirror once per minute for the release of new files, and once a file is selected for measurement, the Getpeer and Peerping scripts are also activated every minute. So we are able to obtain the IP addresses of the peers that injected 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. When a peer is behind a firewall, our Getpeer script can obtain its IP number, but it cannot send any message to it. Firewalls constitute a problem for P2P systems in general [13]. 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 (default 20) of *randomly* selected peer IP numbers. We obtained a sufficiently large list for reliable performance measurements by frequently asking the tracker for a list of 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 %. We retrieved peer IPs at a speed of 60 per minute by starting the Getpeer script from three DAS nodes. Therefore, we obtained a high probability of discovering new peers who just started downloading. Our final measurement problem was caused by modifications made to the BitTorrent system itself. For example, on July 23, 2003, `suprnova.org` changed the way it balances requests across its mirrors,

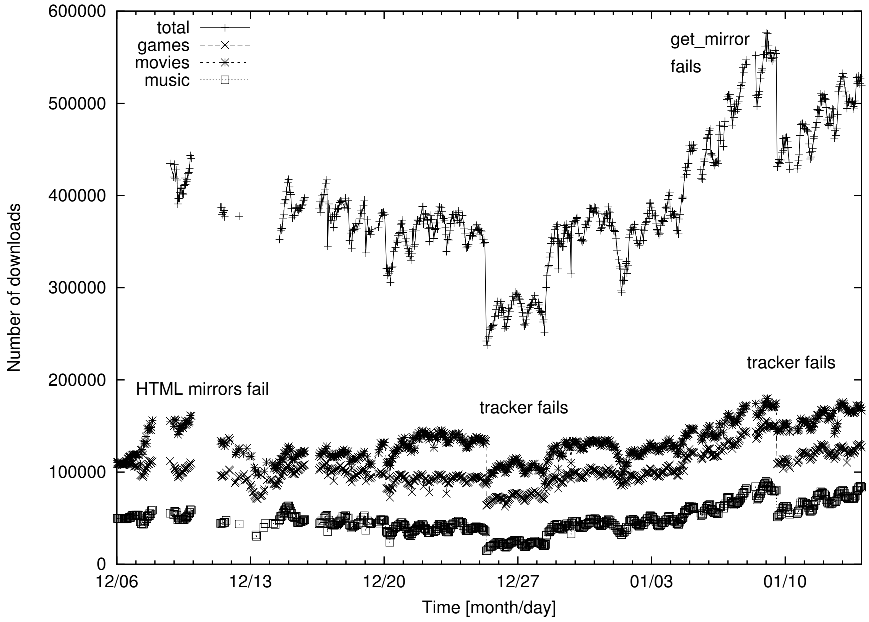


Fig. 3. Number of users downloading or seeding on BitTorrent/Suprnova for one month.

which broke our measurements at that point and created a gap in our traces.

5 Measurement results

In this section we present detailed measurements of BitTorrent for each of the five system characteristics identified in Section 3.1.

5.1 Popularity

This section shows the popularity of BitTorrent/Suprnova in terms of the number of downloads over time and its dependence on technical failures in the system. For many P2P systems it is very difficult to determine this number, but in BitTorrent it is reported by Suprnova.

Figure 3 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 selected this month for presentation because it shows a large variance in the number of downloads due to several BitTorrent/Suprnova failures. The lowest and highest number of downloads in Figure 3 are 237,500 (on Christmas day) and 576,500 (on January 9). Our HTML script requests all pages from one of the active Suprnova mirrors every hour. On the horizontal axis in Figure 3 we use the timestamp embedded in the HTML page indicating when it was generated. The consecutive data points are connected with a line when there was no failure. The total number of downloads is not shown when the pages for a certain type of content could not be retrieved.

There are two things to be noted in Figure 3. The first is the diurnal 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 [6]. The second is the large variation due to failures of either the mirroring system across the Suprnova mirrors, the mirrors themselves, the `.torrent` servers, or the trackers. For example, on December 8 and 10, a gap occurred due to failures of the mirroring system and of 6 out of 8 Suprnova mirrors, and on Christmas day, the large tracker `beowulf.mobilefrenzy.com` 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. The gap around Christmas is actually a few hours smaller than the 98 hours of down time because outdated tracker statistics remain on `suprnova.org` for a few hours when a tracker cannot be reached for an update. On January 9, 2004 again a big tracker went off-line, causing a reduction of 122,000 downloads. From January 5 to 10, the mirroring system was also off-line a few times, causing `suprnova.org` to be unusable and the HTML mirrors not being updated, which is visible in the figure as a few gaps in the total line. The figure suggests that users are not discouraged by such failures.

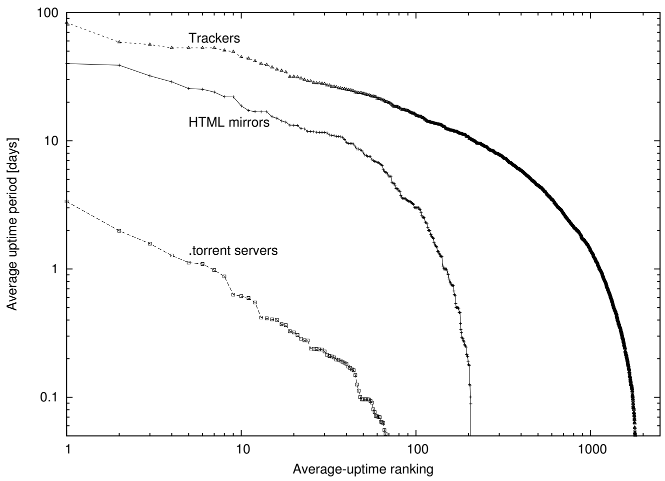


Fig. 4. Uptime ranking of the three BitTorrent/Suprnova central components.

It should be noted that a failure is not always destructive. Users bookmark stable mirrors in order to be independent of the mirroring system, and upon a tracker failure, the downloads in progress can still be completed successfully when a sufficient number of peer IP numbers is known.

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

5.2 Availability

The BitTorrent/Suprnova architecture is vulnerable because of potential failures of the three central 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 750,000 daily visitors, and sometimes, fewer than five mirrors were up. Occasionally, no `.torrent` file servers are available, blocking all new downloads. Trackers are a frequent target for denial-of-service attacks and consume GBytes of bandwidth every day. Suprnova used to operate a few trackers itself, but seems to have stopped doing so due to the high financial and maintenance requirements.

Figure 4 shows the results of our availability measurements of 234 Suprnova mirrors, 95 `.torrent` file servers, and 1,941 BitTorrent trackers. The figure shows their rank based on their average uptime. Only half of the Suprnova mirrors has 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 has an average uptime of 1.5 day or more, and the 100 top ranking trackers have an average uptime of more than 15.7 days.

Figure 3 showed that unavailability has a significant influence on popularity. Combined with the high frequency of such failures, we conclude that there is an obvious need to decentralize the central elements. 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 central components.

5.3 Download performance

Figure 5 compares the number of downloads for a single file (the third Lord of the Rings movie of 1.87 GByte) obtained from three sources, and gives insight into the accuracy of our Peerping script and the Suprnova statistics. We selected this file because it uses the FutureZone.TV tracker 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

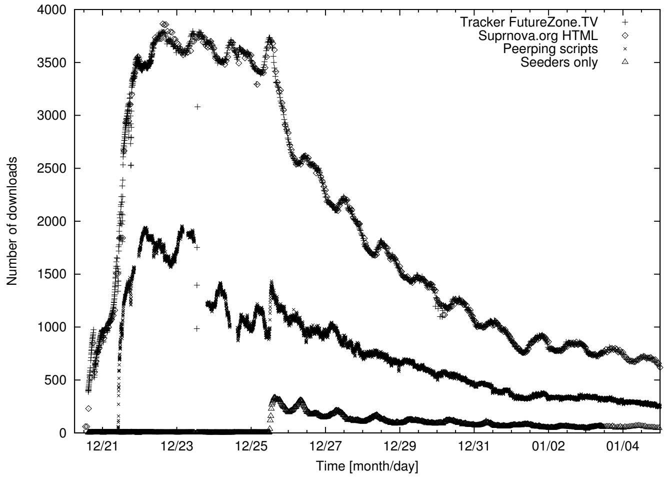


Fig. 5. Comparing three sources for the number of downloads of "Lord of the Rings III".

seeds according to the 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 is continuously online. This long time period provides a clear opportunity to identify copyright violators and to trace their origins. The statistics from Suprnova are fetched by our HTML script every hour, and are in agreement with the total tracker results to such an extent that the lines largely overlap. Only on December 23, 2003 there is 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. Our Getpeer script achieves a sufficient peer coverage, but over 40 % of the peers are behind firewalls. The gaps in the Peerping results are due to disk quota problems on the DAS, which runs the measurement software. The important observation is that the Suprnova statistics agree with the tracker statistics and that the overall trend in the tracker statistics is also reflected in our Peerping results.

Figure 6 presents the results of a two-week experiment in which the bandwidth of thousands of peers was measured. Our Hunt script followed the first 108 files that where added to Suprnova on March 10, 2004. The figure shows the average bandwidth of the 54,845 peers we followed. It turns out that 90% of the peers have a

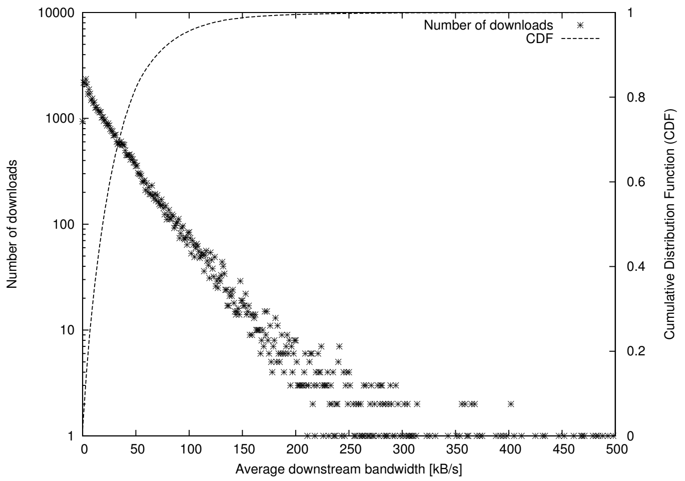


Fig. 6. Average download performance of BitTorrent/Suprnova peers.

download speed below 65 KB/s; the average download speed of 30 KB/s allows peers to fetch even large files in a day. The figure also shows the Cumulative Distribution Function (CDF) of the fraction of peers with a certain download speed. An important observation is the exponential correlation between the average download speed and the number of downloads at that speed.

5.4 Content lifetime

Due to the architecture of BitTorrent the availability of content is unpredictable. Suprnova only provides references to peers with content. Figure 5 shows the flash crowd effect when a file is first injected as well as the gradual decrease in popularity. When the last peer/seed with certain content goes offline, the content dies. In this section we explore the properties of this content lifetime. The dynamics of content lifetime are important because the availability of content is of prime interest to users and therefore an important performance metric.

Figure 7 shows the content lifetime of all large files (at least 500 MBytes) on BitTorrent/Suprnova we have followed between August 2003 and March 2004. Each file is represented as a data point with on the horizontal axis the number of seeds

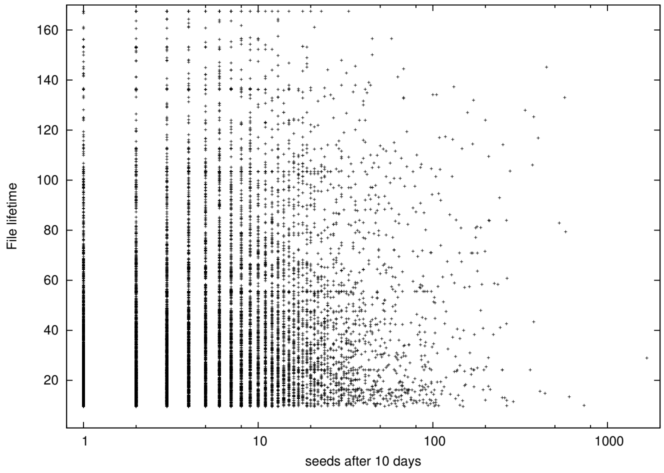


Fig. 7. The content lifetime versus the number of seeds after 10 days for over 14,000 files.

for this 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. Therefore, we shall now examine the seeding behavior in more detail to determine its exact role in content lifetime.

On December 10, 2003 the popular PC game "Beyond Good and Evil" from Ubi Soft 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 accurately traced by our Peerping script. With a 2-minute resolution we measured the uptime of all non-firewalled peers.

Figure 8 shows the results of our uptime measurement. The vertical axis shows the uptime of peers in hours *after* they finished downloading. The horizontal axis shows the individual peers, sorted by 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 is finished. This sharp drop has

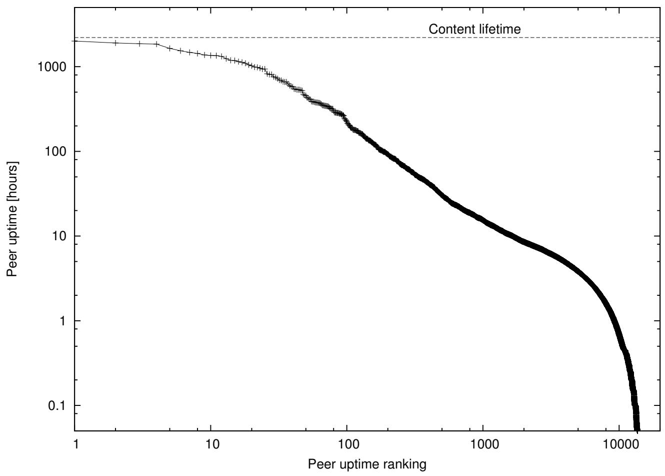


Fig. 8. Uptime distribution for 53,833 peers for a single archive.

important implications because the actual download time spans several days.

Figure 8 shows that seeds with a high uptime 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 %). We observed that this phenomenon of a few highly reliable seeds also applies to other files, and explains why the number of seeds is not a good predictor for content lifetime: having a single peer with a high uptime makes the difference between a content lifetime of 10 days or 100 days. Note that this detailed picture of the uptimes and content lifetimes is only possible because of our detailed measurements over a long period of time.

5.5 Pollution level

The pollution level of BitTorrent/Suprnova content is difficult to measure directly. In order to do so, we could have downloaded all new content during say a week (over 2,500 files on average in March 2004) and manually check the pollution level, but this is far too time consuming. P2P message boards and other sources strongly indicate that BitTorrent/Suprnova is virtually pollution free. Instead of a direct pol-

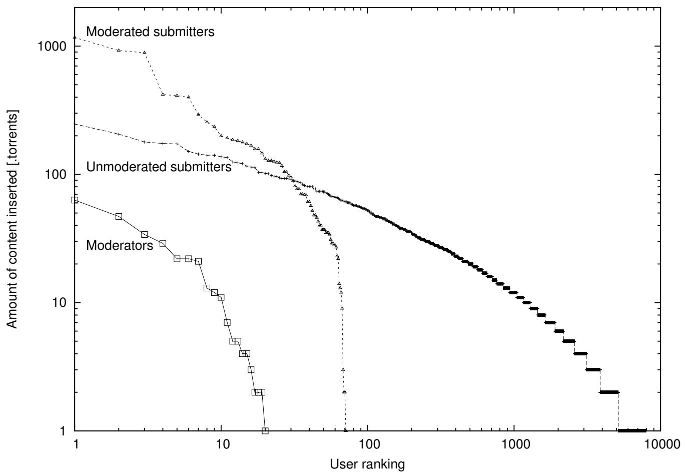


Fig. 9. The activity of the different volunteers on Suprnova to prevent pollution.

lution level measurement we show the performance of the P2P moderation system.

Figure 9 shows the numbers of files that are injected by the 18 moderators, the 71 unmoderated submitters, and the 7,933 moderated submitters that where 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 were surprised that a mere 18 moderators are able to effectively manage the numerous daily content injections with such a simple system. Unfortunately, this system relies on a central server and is extremely difficult to distribute.

6 BitTorrent improvements

In this section we discuss two improvements to BitTorrent/Suprnova. The first concerns the fact that BitTorrent/Suprnova is only partially distributed, with its central elements being sources of *availability problems*. The only thing that BitTorrent peers share is the actual content. This situation can be improved making the peers also share indexes of content, the content integrity checksums, and peer IP numbers. This requires an infrastructure for trust, reputation management, and message validation to prevent malicious peers from spreading false information. Fortunately, this can be accomplished by using public/private keys, Merkle hash trees, and other techniques [14]. The functionality of `suprnova.org` may then be reduced to the validation of correct content and moderators using public/private keys. By using Merkle hash trees, a single SHA1 value can be used to check the validity of content, indexes, and peer IP numbers. A gossip-style protocol is suited for the actual distribution [15] of this control information.

The second improvement concerns the *download speed* and the *content lifetime*, which can benefit from better cooperation and network-awareness. The download speed can be enhanced by having peers keep track of performance information of past download speeds, hopcounts, latencies, and the uptimes of peers, and by making them share and exploit this information. Currently, BitTorrent still has to use trial-and-error to select peers for bartering in a flash crowd with thousands of peers. Failing to exploit peers with exceptionally long uptimes is another missed opportunity, as they can be used to increase the availability of the performance information. Of course, the exchange of this information entails some overhead. Content lifetime can be improved dramatically if sharing of old content is rewarded. Currently, BitTorrent's tit-for-tat mechanism limits bartering between two peers to a single file at any point in time. Microcredits are needed to make this system more general.

7 Discussion and conclusions

In this paper we have presented a measurement study of the BitTorrent/Suprnova P2P system, where we considered five characteristics: the popularity, the availability, the download speed, the content lifetime and injection time, and the pollution level. This constitutes the longest and most comprehensive measurement study of P2P systems to date, and we believe that it 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), measurements software, and documentation on `peer-2-peer.org`.

Our popularity measurements show that the number of downloads in BitTorrent/Suprnova is strongly influenced by the availability of the central components. We concluded that the lack of decentralization in BitTorrent/Suprnova is the cause of the availability problems, and we proposed an improvement to the system by completely decentralizing the functionality of the central components across the peers in order to solve the availability problems.

Our measurements of the uptimes of peers show that some of these uptimes are very large (e.g., of the 53,000 peers we tracked for several months, a fraction of <0.4 % have an uptime higher than 100 hours; to measure this accurately, both the measurement duration and number of peers need to be sufficiently large). We concluded that peers should not be used indiscriminately, but that it might be possible to significantly improve the download speed and the content lifetime by having peers exchange performance information.

The increasing use of firewalls and NATs with their diversity and unpredictable behavior are a growing problem for P2P systems because they decrease the download speed. Little is known about them in relation to P2P systems, and topics such as firewall awareness, TCP over UDP tunneling, and rendezvous points require more research. A final important observation is that content creators suffer large revenue losses due to P2P systems; we view copyright violations as the main obstacle for the future of P2P systems. It should be investigated whether initiatives such as the

Apple Itunes music store can somehow be merged with the P2P concept to solve this problem.

8 Acknowledgements

The authors would like to thank Maarten van Steen from the Vrije Universiteit Amsterdam for his valuable comments on the draft versions of this paper. The first author would also like to thank Frans Kaashoek from MIT for the opportunity of working with his group and for all help during the initial experimental setup.

References

[1] B. Cohen, "Incentives Build Robustness in BitTorrent", `bitconjurer.org/BitTorrent`, May 2003.

[2] E. Adar, B. Huberman, "Free riding on Gnutella", Technical report, Xerox PARC, 10 Aug. 2000.

[3] P. Backx, T. Wauters, B. Dhoedt, P. Demeester, "A comparison of peer-to-peer architectures", Eurescom Summit 2002, Heidelberg, Germany.

[4] N. Leibowitz, M. Ripeanu, A. Wierzbicki, "Deconstructing the Kazaa Network", 3rd IEEE Workshop on Internet Applications (WIAPP'03), June 23-24, 2003, San Jose, CA.

[5] R. Bhagwan, S. Savage, G. M. Voelker, "Understanding Availability", Int. Workshop on Peer to Peer Systems, Berkeley, CA, Feb. 2003.

[6] S. Saroiu, P. K. Gummadi, S. D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems", Multimedia Computing and Networking 2002 (MMCN '02).

[7] K. P. Gummadi et al., "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload", 19-th ACM Symposium on Operating Systems Principles (SOSP'03), Oct. 2003.

[8] S. Byers, L. Cranor, E. Cronin, D. Kormann, P. McDaniel, “Analysis of Security Vulnerabilities in the Movie Production and Distribution Process”, The 2003 ACM Workshop on DRM, Oct. 2003.

[9] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, M. Faloutsos, “File-sharing in the Internet: A characterization of P2P traffic in the backbone”, Technical Report, UC Riverside, 2003.

[10] J. Chu, K. Labonte, and B. Levine, “Availability and locality measurements of peer-to-peer file systems”, ITCom: Scalability and Traffic Control in IP Networks, July 2002.

[11] S. Sen and J. Wang, “Analyzing Peer-to-Peer Traffic Across Large Networks”, ACM/IEEE Transactions on Networking, Vol. 12, No. 2, April 2004, pp. 137–150.

[12] C. Nieman, “Digital Decoys”, IEEE Spectrum, May 2003, p. 27.

[13] T. Oh-ishi, K. Sakai, T. Iwata, A. Kurokawa, “The deployment of cache servers in P2P networks for improved performance in content-delivery”, Third International Conference on Peer-to-Peer Computing (P2P’03), Sep. 2003.

[14] G. Nuckolls, C. Martel, S. G. Stubblebine, “Certifying Data from Multiple Sources Submitted for publication”, `truthsayer.cs.ucdavis.edu`

[15] P. Eugster, R. Guerraoui, “Probabilistic Multicast”, IEEE Symposium on Dependable Systems and Networks, Washington DC, June 2002.