BTWorld: Towards Observing the Global BitTorrent File-Sharing Network

Maciej Wojciechowski
TU Delft, the Netherlands
M.Wojciechowski@tudelft.nl

Mihai Capotă
TU Delft, the Netherlands
M.Capota@tudelft.nl

Johan Pouwelse
TU Delft, the Netherlands
J.A.Pouwelse@tudelft.nl

Alexandru Iosup
TU Delft, the Netherlands
A.Iosup@tudelft.nl

ABSTRACT

Today, the BitTorrent Peer-to-Peer file-sharing network is one of the largest Internet applications—it generates massive traffic volumes, it is deployed in thousands of independent communities, and it serves millions of unique users world-wide. Despite a large number of empirical and theoretical studies, observing the state of the global BitTorrent network remains a grand challenge for the BitTorrent community. To address this challenge, in this work we introduce BT-World, an architecture for observing the global BitTorrent network without help from the ISPs. We design BTWorld around three main features specific to BitTorrent measurements. First, our architecture is able to find public trackers, that is, the BitTorrent components that offer unrestricted service to peers around the world. Second, by observing the state of these trackers, BTWorld obtains information about the performance, scalability, and reliability of BitTorrent. Third, BTWorld is designed to pre-process the large volumes of recorded data for later analysis. We demonstrate the viability of our architecture by deploying it in practice, to observe and analyze one week of operation of a large part of the global BitTorrent network–over 10 million swarms and tens of millions of concurrent users. We also show that BT-World can shed light on BitTorrent phenomena, such as the presence of spam trackers and giant swarms.

Categories and Subject Descriptors

C.1.4 [**Parallel Architectures**]: Distributed architectures; C.4 [**Performance of Systems**]: Performance attributes; D.4.8 [**Performance**]: Measurements

General Terms

Performance, Measurement, Experimentation

1. INTRODUCTION

BitTorrent is a peer-to-peer, file-sharing, Internet-based application that has experienced phenomenal community growth in the past five years. Due to a variety of factors, such as robustness to content pollution [25] and good performance [13,15,25], BitTorrent deployments—of different community size and shared content type—currently serve tens of millions of daily users. However, the growth of BitTorrent into a global file-sharing network has begun to raise similar issues as the Internet did a decade ago [3], among which the strategic importance of trace-driven research and development. Many empirical studies [13, 15, 25, 30] have been dedicated to collecting traces from and to understanding the properties of BitTorrent deployments, but these studies have been limited in size and have thus introduced various sampling biases [34]. Complementing these studies, in this work we introduce BTWorld, which is an architecture for observing and analyzing the *global* BitTorrent network, and we use it to characterize BitTorrent deployments world-wide.

Measuring the global BitTorrent network is long overdue. Indeed, the need for new data is not unique to BitTorrent among Internet applications. The research community can greatly benefit from usage traces taken from real applications, for example to understand the trends and characteristics of real deployments, to validate models and drive simulations, and, ultimately, to improve the quality of service for millions of users. Motivated by the wide applicability of data, the US National Academy of Sciences has challenged the Internet research community to *develop the means to capture a day in the life of the Internet* in 2001 [3], and gathered the data collected by small independent measurements into public archives such as DatCat [29] and CRAW-DAD [33]. While BitTorrent has grown to become a significant part of the Internet, and the BitTorrent research community has produced numerous measurement studies, as a community we have not yet produced a coherent view of the global BitTorrent network, that is, we have not yet formulated our "one day in the life of ..." grand challenge and we have not created a public archive with our traces. What should be included in a global BitTorrent view? To guide the development of the network we envision obtaining, from a statistically-significant set of BitTorrent deployments, information about BitTorrent's performance, scale, and reliability of service.

The problem of observing the global BitTorrent network is not trivial. While large Internet studies are enabled by cooperation with the ISPs, for BitTorrent few ISPs will agree to observing the network, for fear of legal consequences; for example, the RIAA has subpoenaed ISPs to reveal the identities of customers involved in alleged copyright infringements [20]. Thus, the BitTorrent community has been forced to perform its own measurements, without assistance from major ISPs. Due to the scale, complexity, and rapid growth

HPDC'10, June 20–25, 2010, Chicago, Illinois, USA.

of the network, the results of numerous previous BitTorrent measurements are difficult to interpret coherently. It is symptomatic for the current (lack of) understanding of the current global status of BitTorrent that there is no consensus on the Internet traffic share generated by BitTorrent users–though caching companies have proposed estimates of 30% in 2005 [22] and over 50% in 2008 [14]. Moreover, academic studies in consecutive years report the doubling of the average download capacity [13,25], studies spread five years apart of the same BitTorrent deployments acknowledge significant network changes [34], etc. Furthermore, the existing BitTorrent studies are made on a changing network: while the BitTorrent protocol has remained virtually unchanged since its introduction, a number of additional protocols have made the network less centralized, more available, and more difficult to observe (see Section 2.1).

In this work, our main contributions toward a system for observing the global status of BitTorrent are:

1. We formulate the requirements for the problem of observing the global status of BitTorrent (in Section 2);
2. We design the BTWorld architecture for observing the global status of BitTorrent (in Section 3);
3. We implement BTWorld and use the resulting system to observe and analyze one week of operation in the life of a large part of the BitTorrent network (in Section 4);
4. We show that the wealth of data collected by BTWorld can shed new light on BitTorrent phenomena such as spam trackers and giant swarms (in Section 5).

2. THE PROBLEM

In this section we formulate *the problem of observing the global BitTorrent network*. We first introduce BitTorrent, then the system requirements for addressesing the problem.

2.1 The BitTorrent File-Sharing Network

BitTorrent is a peer-to-peer file-sharing system for exchanging *content* in the form of (bundles of) files. Some BitTorrent file-sharing is *private* [19], with access restricted to access to a group of authenticated users. In this paper, we focus only on *public* BitTorrent file-sharing activities.

In our terminology [34], BitTorrent has three levels of operation, the peer level, the swarm level, and the community level, which we describe in turn. The *peer level* comprises individual peers, that is, individual participants in the process of exchanging content that may contribute to or use the system's resources and services. Peers are present in the system during non-contiguous intervals called sessions; the system size at any moment in time is given by the number of sessions overlapping at that moment in time. The *swarm level* groups together, from all the peers in a P2P system, the peers that exchange the same content, for example the latest Linux distribution [15]. A swarm starts being active (is born) when the first peer joins that swarm, and ends its activity (dies) when its last peer leaves. The lifetime of a swarm is the period between the start and the end of the swarm; the swarm is *alive* during its lifetime. The *community level* comprises all the swarms and all the peers in a BitTorrent deployment such as SuprNova [25], ThePirateBay [13,30], E-tree [1], etc. The main function of the community level is to facilitate the formation of swarms, that is, to allow the peers interested in exchanging the same content to start collaborating. This collaboration is facilitated by a *tracker*, which is a specialized system component that operates at both the community and the swarm levels, and acts as a meeting point for peers. A tracker can manage several swarms.

Table 1: Major changes in BitTorrent.

Change (Year)	Effect
Multitracker Metadata (2003)	higher tracker availability
DHT Protocol (2005)	global BitTorrent network
Peer EXchange (2007)	trackerless BitTorrent
P2P Metadata (2008)	webless BitTorrent search

The content transferred in BitTorrent is accompanied by a *.torrent* metadata file. The metadata includes, among other pieces of information, a *hash* that uniquely identifies the content, and a directory for the trackers that manage swarms exchanging the content. Peers interested in a file obtain the file's metadata from a web site (the community level of BitTorrent) and use the peer location services offered by a tracker (the community and swarm levels of BitTorrent) to find other peers interested in sharing the file. The raw file is then exchanged between peers (the peer level of BitTorrent). Thus, to obtain a file, a user partakes in all three operational levels of BitTorrent.

Several important shortcomings in the original BitTorrent design have been addressed over time, and four main changes are summarized in Table 1. First, the directory part of the metadata was initially only able to specify a single tracker, which raised an availability problem; the Multitracker Metadata Extension [12] introduces support for multiple trackers. Second, peers were able to discover other peers only through the services of trackers, which raised anonymity concerns; the DHT Protocol [17] introduces support for finding and exchanging information directly between peers and creates a coarse global BitTorrent network. Third, even when connected through the DHT into a global BitTorrent network, peers had no protocol to inform each other of the presence of other peers in a swarm; the Peer EXchange (PEX) protocol [6] implements this feature, thus enabling swarms to function trackerless. Fourth, to be able to exchange content peers needed to be able to know the content metadata, which was only obtainable from BitTorrent (web) sites. With the Extension for Peers to Send Metadata Files [11], peers can send to each other metadata, effectively creating a webless BitTorrent network.

2.2 System Requirements

The global BitTorrent network comprises all the BitTorrent communities in the world. Thus, the problem of observing the global BitTorrent network is, in a first formulation, the problem of building a system capable of observing and recording the complete state (traffic, connections, and participation at different BitTorrent levels) for all the public BitTorrent communities in the world. A similar problem has been proposed by the general networking research community as "a day in the life of the Internet"[1], but with a focus on network instead of application operation.

The first formulation of the problem is the expression of an archival effort, which would require observing and recording

[1] A paraphrase on Solzhenitsyn's book *One Day in the Life of Ivan Denisovich*, which depicted the horrors of the Soviet labor camps in the 1950s by recounting the details of a day spent in a camp by the eponymous prisoner.

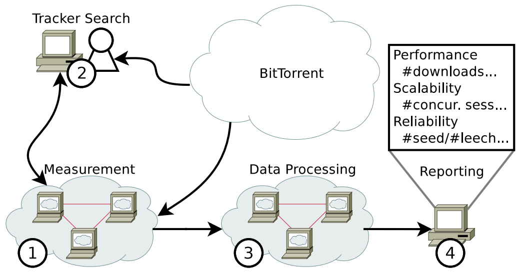

Figure 1: The architecture of BTWorld.

everything in the BitTorrent network. However, our goal is more modest: to reveal the performance, scalability, and reliability of the global BitTorrent network. This allows us to make the problem tractable by focusing on BitTorrent trackers, and not each individual peer.

Besides the good properties of any measurement system, that is, fault-tolerance, scalability, and non-intrusiveness, we identify three system requirements specific to our problem:

1. The system must be able to find the important trackers serving public BitTorrent communities. Since most BitTorrent trackers are short-lived [21, 25], we define the *important trackers* as the long-lived trackers that serve either large communities or communities with special characteristics (e.g., religious content);
2. The system must be able to observe and record information about the performance, the scalability, and the reliability of the important trackers. To increase system reliability, some of the trackers have been replicated; the system should observe only one instance of each replicated set of trackers;
3. The system must be able to facilitate data analysis. The volumes expected for the observation data may easily run into the tens of terabytes per year. Moreover, the observation process may take days or even weeks before accumulating sufficient data for the problem at hand, so timely detection of errors is critical in the observation process. Thus, the system must provide periodical summaries of the recorded data.

3. THE DESIGN OF BTWORLD

In this section we present the design of BTWorld, which follows the system requirements introduced in Section 2.

3.1 Overview

Our goal is to create a global view of the publicly accessible BitTorrent network that will enable us to understand system characteristics such as performance and scalability, and to identify and characterize BitTorrent phenomena. We design this architecture around data observed from trackers (see Section 3.2).

A distinguishing feature of the BTWorld architecture is its focus on low resource usage. BTWorld needs to balance two conflicting requirements: scaling to the global size of BitTorrent and keeping the amount of consumed resources (such as machines and IP addresses) low. We specifically want to use an order of magnitude fewer machines than previous infrastructures [13,30], which have each employed several hundred machines; we use only two machines (eight cores) to perform a large-scale BitTorrent measurement and process its results (see Section 4), and to observe and analyze new BitTorrent phenomena (see Section 5).

Table 2: Measurement techniques.

Level	Advantage	Disadvantage
Internet	Excellent coverage	ISP collaboration
Community	Implementation	Peer details
Swarm	Details	Context
Peer	Details	Scalability

Figure 1 shows the four BTWorld components and the data flows between them. The *Measurement* component (component 1) connects to trackers in the public BitTorrent network and collects their status (the raw data). The *Tracker Search* component (2) gathers information about new trackers, and produces a list of the most important trackers based on various sources of information. The BT-World design specifically permits domain experts to configure and guide the tracker selection process. The Measurement component contacts *only* the most important trackers (see Section 3.3), which are provided by the Tracker Search component. The *Data Processing* component (3) creates statistics from the raw data. Last, the *Reporting* component (4) transforms the output of the Data Processing component into useful reports, including tables, graphs, and animations.

3.2 Measurement Techniques

While designing BTWorld, we have identified four different measurement techniques, distinguished by the level at which the measurement is performed. We describe in the following each of these techniques and summarize their main advantages and disadvantages in Table 2.

Internet-level measurements involve monitoring the traffic exchanged between peers, for example by observing the traffic passing through the routers of an ISP [16, 28]. While this approach can result in observing the complete BitTorrent activity (perfect coverage), it also requires access to the routers of many large ISPs. Moreover, interpreting the traffic is difficult due to the use of encryption in BitTorrent.

The next option we have considered is to observe BitTorrent at the *community level*. The passive measurement technique of observing the logs produced by trackers [15] leads to good measurements, but requires the collaboration of community administrators, which raises legal concerns. We opt instead to use the scrape service available on most of the trackers. The scrape service is a lightweight protocol running over HTTP, which provides swarm status information to users. It includes, for each swarm, the numbers of seeders, leechers, and downloads.

In contrast to community-level measurements, *swarm-level* measurements focus on individual swarms. They can also use the scrape service, but query information for a single swarm [13, 25]. In addition, they can collect information through PEX [30]. Swarm-level measurements lack the context information provided by community-level measurements, e.g., multi-swarm participation of users [9].

Peer-level measurements rely on communicating with peers using a custom BitTorrent client [7, 13], which can can log activity and even deviate from the strict BitTorrent protocol [7,13]. It is difficult to deploy this approach at the global scale required by BTWorld.

Considering the requirements in Section 2.2, we base our

Table 3: Tracker search techniques.

Technique	Advantage	Disadvantage
Well-known sites	Implementation	Coverage
Pre-compiled lists	Niche sites	Reliability
Parsing torrents	Exploit data	Data size
Metadata extension	Hash analysis	Success rate
Web search engines	Unknown sites	Implementation

architecture on community-level measurements performing tracker scrapes and web site scrapes. Additionally, we employ *torrent file retrieval* – obtaining the torrent file corresponding to a hash observed in a tracker scrape. We have a twofold objective for doing this: improving our tracker knowledge with multi-tracker torrents as we show in the next subsection, and determining the size of the content seen in tracker scrapes. However, torrent file retrieval incurs a significant overhead for millions of torrents – to obtain file size information it collects full torrent files, sized 30-100 kB each. Instead, we use custom scripts that parse the directory pages of well-known BitTorrent sites such as `ThePirateBay.org`, `Kickasstorrents.com`, and `YourBitTorrent.com`, and extract content size information.

The raw data that we collect includes the hashes identifying shared content in each community, the number of peers participating in swarms (both as seeders and as leechers), the number of downloads for each swarm and the size of the content being shared.

3.3 Tracker Search

The Tracker Search component selects the trackers observed during measurements. We have identified five techniques for finding BitTorrent trackers. We describe in the following each of these techniques and summarize their main advantages and disadvantages in Table 3.

The *Well-known BitTorrent sites* technique uses common knowledge to select trackers. For example, we have selected SuprNova in 2004 [25] and PirateBay in 2005 [13] as the largest from a set of well-known BitTorrent sites; Andrade et al. [1] select one well-known site among the three used in their study; etc. Such an approach requires domain expertise, does not react to changes in the status of the sites (SuprNova is no longer online, PirateBay was shut down several times for legal reasons), and may disregard small BitTorrent sites that host important trackers.

Lists of trackers are publicly accessible on the web. The largest list we know about is located at `p2ptrace.com`; previous studies have used `torrentspy.com` (now defunct). Other lists use expert knowledge to include sites focused on niche areas (e.g., Bollywood movies, Japanese manga, etc.). Reliability is a concern with this solution, since unmaintained lists include a quickly increasing number of dead trackers. `p2ptrace.com` traces about 900 trackers and is actively updated, but has a complex presentation format and its server is often overloaded.

Parsing torrent files–Most of the torrent files we encountered use multiple trackers. We extend our tracker coverage by parsing the torrent files collected through the other methods, and selecting from them the trackers that appear often.

The BitTorrent metadata sending extension enables us to obtain torrent files for parsing using the hashes observed in scrapes. BTWorld implements this functionality as an automated process built on top of a BitTorrent client.

Web search engines provide an alternative way to collect torrent files. Specialized search engines such as `torrentz.com` allow us to retrieve torrents corresponding to specific hashes, but limit the transfer rate offered to each IP and are thus very slow. Alternatively, a custom search on Google ("filetype:torrent") returns about 5.5 million results at the time of writing. Parsing these files can extend the coverage of BTWorld, but requires effort to retrieve and integrate data.

We have implemented and employed in BTWorld all these techniques, and found empirically that using them simultaneously increases the coverage of the tracker search.

3.4 Data Processing

BTWorld generates large amounts of data, and processing may build in-memory working sets on the order of gigabytes. Thus, it is impractical to process the raw measurement data once for every statistics output. We are thus facing the problem of using a MapReduce-like approach vs. a Parallel DBMS [23]. With a MapReduce-like approach, the raw measurement data are first pre-processed into intermediate data, which are then used as input to several statistics. With a Parallel DBMS, the raw measurement data are loaded into the database, and then the execution queries for the different statistics are optimized by the database engine. The database community has recently observed [23] that the selection of either of these two alternatives depends on the parameters of the data processing problem (data size, duration of data loading, complexity of processing operations, etc.). We perform our own experiments with the dataset described in Section 4.1 and found that, for our problem and experimental setup, the MapReduce-like approach produces better results than the parallel DBMS; we are currently performing a further investigation of this topic.

3.5 BitTorrent Analysis

We aim to analyze BitTorrent in terms of global system characteristics and unique phenomena, both qualitative and quantitative. For the quantitative analysis, we use the following metrics, grouped by system characteristic:

Performance - to quantify system performance, we look at the total number of downloads per swarm. We also consider the total downloaded volume per swarm, defined for a swarm as the product of its content size and its total number of downloads. We further define the total number of downloads per tracker as the sum of the number of downloads for all the swarms managed by the tracker, and the total downloaded volume per tracker as the sum of the downloaded volume of the swarms managed by the tracker. We also present the number of downloads per hour for a tracker defined as the difference in reported number of downloads for all the swarms of the tracker in two samples taken one hour apart.

Scalability - to analyze scalability, we define the following two metrics: number of concurrent sessions and content size. The number of concurrent sessions per swarm is the sum of the number of seeders and leechers present concurrently in a swarm, as reported by the tracker in the scrape. We further define the number of concurrent sessions per tracker as the sum of the number of concurrent sessions for every swarm managed by the tracker. The content size for a swarm is the size appearing in the torrent file corresponding to the swarm. The content size for a tracker is the sum of the content swarm for every swarm managed by that tracker.

Table 4: Characteristics of the collected traces.

Metric	Value
Tracing Period	Jan 03-09, 2010
Number of trackers	912
Number of alive trackers	769
Number of swarms	10,329,950
Number of hashes	6,314,318
Number of hashes with known size	1,024,573 (16.2%)
Number of swarm samples	899,537,250

Reliability - we express reliability by looking at the number of dead torrents, defined as the number of torrents with no seeders reported by the tracker in the scrape. We also investigate the seeders to leechers ratio, which is the ratio of the number of seeders to the number of leechers in that swarm, as reported by the tracker.

4. A WEEK IN THE LIFE OF 769 BITTORRENT TRACKERS

In this section we analyze a large-scale BitTorrent dataset collected by BTWorld over one week of operation.

4.1 BitTorrent Traces

We started collecting traces using BTWorld in December 2009, and the collection process still continues. For this work we have selected a one week subset of the entire trace collection, starting on January 3, 2010. January is typically a slow month in user content distribution, especially in comparison with the usual peak loads occurring in December; many of the users in US and Europe (the majority of the users of MiniNova, PirateBay [13], and other large sites focusing on English-speaking content) return from vacations and start the work year. This dataset serves two purposes: as an assessment of the BTWorld capabilities, and to show the usefulness of the raw data acquired so far by BTWorld.

BTWorld was able to collect in one week a large-scale dataset, as summarized by Table 4. The total number of swarms present in our traces exceeds 10 million, and overall we have collected close to 900 million swarm samples with swarm information (number of seeders and leechers, etc.) The data processing for this dataset took less than 7 hours to complete: 4 for preprocessing the raw data, 2 hours for generating the intermediary files, and a few minutes for generating all the tables and graphs. While further optimizations in the data processing workflow are possible, we consider the current time requirements acceptable.

The collected data reveal interesting features of the global BitTorrent network; we comment in the following on two such features. We identify *alive trackers* as those that replied at least once with a meaningful response (i.e., we were able to parse the response and produce useful data). Out of our initial list of trackers, only 769 (84%) were alive during the selected week; this re-confirms the findings of previous tracker availability studies [21, 25]. The total number of swarms is 1.63 times the number of hashes, indicating significant overlaps in the content of different trackers.

4.2 Global Analysis

We first analyze the trace from a global perspective, trying to identify characteristics of the BitTorrent network as a whole. We exclude here the spam trackers presented in detail in Section 5.1; these trackers return, via the scrape interface, information that cannot be trusted.

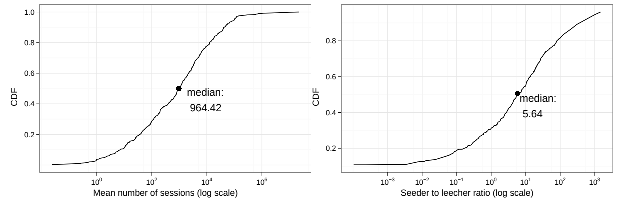

Figure 2: CDF of the average number of sessions (left) and seeder to leecher ratio (right) per tracker.

Large diversity in tracker sizes. We investigate the number of users served by individual BitTorrent trackers. Figure 2 (left) shows the cumulative distribution function (CDF) of the number of concurrent sessions for all the trackers. We observe in the figure and in Table 5 a wide range of values, between 75 and 27.6 million concurrent sessions. The median of around 1,000 concurrent sessions shows that most BitTorrent trackers investigated are relatively small compared to the values reported in previous work [9, 13, 15, 25].

High proportion of seeders. We analyze the availability of content in the swarms by looking at the seeding levels per tracker. Figure 2 (right) shows the CDF of the seeders to leechers ratio (SLR) for each tracker. We observe a median SLR of 5.64, which we find high. For 406 (almost 70%) trackers we observe a seeders:leechers ratio above 1. This shows that currently public BitTorrent trackers exhibit high content availability, and the "few seeders many leechers" scenario is not common anymore. The result is in accordance with another recent study, which has investigated private trackers [19], and shows that BitTorrent is overall a well-provisioned network.

BitTorrent users share petabytes (PB) of unique content, mostly in large bundles of files. We focus on the sizes of the (bundles of) files shared, and study the content of 2.29 million swarms; Table 6 summarizes their basic statistics. The total size of the content shared by these 2.29 million swarms is 3.43 PB; assuming that all the 10.33 million swarms observed in our study have similar properties, we conclude that the BitTorrent network comprises at least 15 PB of content. The mean content item size is 1.50 GB, which shows that BitTorrent is mostly used for exchanging large (bundles of) files.

The distribution of the studied characteristics, including the average number of concurrent sessions per tracker and the seeders:leechers ratio per tracker, show elements of exponential probability distribution in the bulk of the distribution, but have significant non-exponential features in the tails. For example, the CDF of the average number of sessions, depicted in Figure 2 (left), shows elements of an exponential distribution between its 7th and 93rd percentiles. However, we notice that for the 93rd-100th percentile the exponential distribution assumption does not hold–there are trackers with more than 20 million concurrent sessions, an order of magnitude larger than the next smaller trackers.

4.3 Per-Tracker Analysis

To present tracker-level details, we first rank the trackers according to the average number of sessions per swarm. We then present in this section the characteristics of the largest ten trackers. We use the same color (for color prints or online access to the article) and symbol to depict the same tracker in Figure 3.

Table 5: Number of sessions: Basic statistics.

	Trackers			
	All	First	Top-10	Top-100
Min	0	17.98 M	116,766	0
Q1	75	19.16 M	591,669	31,420
Median	964	21.91 M	773,222	54,982
Mean	154,300	22.15 M	5.36 M	893,445
Q3	7,077	25.03 M	8.70 M	122,630
Max	27.60 M	27.60 M	27.60 M	27.60 M
StdDev	1.48 M	2.97 M	7.35 M	3.49 M
Sum	7.53 bn	3.70 bn	7.01 bn	7.44 bn

Table 6: File sizes: Basic statistics.

	Trackers			
	All	First	Top-10	Top-100
Min	13 B	1.47 GB	186,368 B	13 B
Q1 B	61.17 MB	1.48 GB	1.75 GB	243.81 MB
Median	366.77 MB	1.49 GB	5.69 GB	735.42 MB
Mean	1.50 GB	20.46 GB	14.27 GB	3.01 GB
Q3 B	1.05 GB	21.79 GB	12.00 GB	2.67 GB
Max	423.55 GB	101.06 GB	329.13 GB	423.55 GB
StdDev	5.18 bn	34.18 bn	26.12 bn	9.40 bn
Sum	3.43 PB	225.09 GB	10.08 TB	716.02 TB

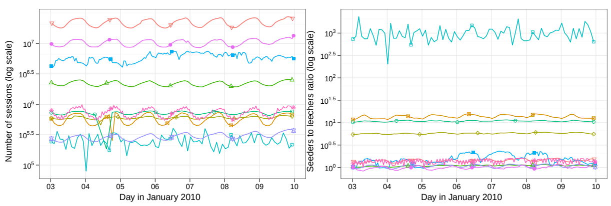

Figure 3: Number of sessions (left), and seeder to leecher ratio (right) in time for the Top-10 trackers.

Daily usage pattern. We focus on the number of users a tracker is serving. Figure 3 (left) shows the number of concurrent sessions for the top ten trackers with respect to time. For almost all the trackers the daily usage pattern is visible. This seems to be a result of human activity, as there exists a strong day-night variation in the recorded data.

Relatively constant seeders to leechers ratio. We study how the content availability evolves in time. Figure 3 (right) shows the seeders to leechers ratio for the top ten trackers, with respect to time. Apart from one tracker, we do not observe large fluctuations in the ratio. Content availability is relatively constant over time, with some daily usage patterns and minor fluctuations visible.

5. NEW BITTORRENT PHENOMENA

In this section we show evidence that BTWorld enables the study of new BitTorrent phenomena. In the following, we study two such phenomena, namely spam trackers and giant swarms, using the traces presented in Section 4.1.

5.1 Spam Trackers

Web spamming is understood as the process of misleading web search engines into giving a web site a rank that is unjustifiably high for the actual content [10]. Similarly, we define *spam trackers* as trackers that advertise false information about swarms for the purpose of spamming specialized BitTorrent search engines. There are multiple reasons for maintaining spamming trackers, such as disrupting the illegal sharing of copyright-protected works, or spreading malicious software; we do not investigate these, but focus instead on the quantitative characterization of the phenomenon.

We propose a rule based system for identifying spam trackers. Currently, the domain expert generates the rules. Our system is based on three rules (criteria). Criterion **C1** defines a spam tracker as a tracker advertising an improbably high number of peers participating in the swarms it manages; the current cut-off point is 50 million. Criterion **C2** considers the average number of downloads per swarm for all the swarms of a tracker; if this number is at any time higher than 1 million, the tracker is spamming. Finally, we define criterion **C3** using the coefficient of variation (CV) of the number of concurrent sessions. Based on the collected data, we empirically set the cut-off point for criterion C3 at 2. All known trackers are well below the cut-off points for all criteria, thus false-positives are not an issue. We will investigate false-negatives in future work.

Table 7: Spam Trackers: Number and percentage of trackers found by criterion and mixtures of criteria. Percentage computed from the number of Alive Trackers (see Table 4).

	Found Spam Trackers	
Applied Criteria	Number	Percentage
C1, $TSessN \geq 50Mpeers$	83	11%
C2, $mean(SDldN) \geq 1Mdownloads$	7	1%
C3, $CV(TSessN) \leq 2$	41	5%
C1 or C2	89	12%
C1 or C3	124	16%
C2 or C3	48	6%
Total (C1 or C2 or C3)	130	17%

Table 8: Largest swarms: Basic statistics for the number of concurrent sessions per swarm.

	Top-1%		Top-0.1%		Top-0.01%	
Samples	15,711		1,571		157	
SSessN	Max	Mean	Max	Mean	Max	Mean
Min	546	9	4,301	255	31,490	9,055
Q1	708	448	7,292	2,088	33,680	15,150
Median	1,025	643	8,185	4,483	36,080	18,860
Mean	2,838	1,509	15,850	7,538	48,920	28,100
Q3	2,046	1,160	17,040	8,147	45,270	19,460
Max	373K	366K	373K	366K	373K	366K

A summary of the spam levels resulting from applying these criteria is presented in Table 7. We observe that most of the spam trackers advertise a very large number of peers, and are thus identified by criterion C1. A considerable number of trackers advertise lowly varying number of peers, and are identified by criterion C3. Last, there exist seven trackers that try to spam using false download numbers (C2). Overall, we notice a lower percentage of spamming compared to other areas–less than 20% vs. 50–80% for email and the Web– but our spam identification methods are still coarse in comparison with the state-of-the-art [10].

5.2 Giant Swarms

Previous studies [15, 25] identify at most 5000 concurrent peers in the largest swarms observed. We classify swarms with a larger number of concurrent sessions as *giant swarms*.

To analyze swarm sizes, we first introduce the notion of *active* swarms as swarms with the number of concurrent sessions above 10 at any time in their operation. Eliminating inactive swarms prunes the data set of information irrelevant for P2P transfers. The spam trackers identified in the previous section are also excluded from the dataset used for analyzing giant swarms.

We now investigate the characteristics of the largest swarms present in our dataset. Table 8 summarizes the basic statistical properties of the number of concurrent sessions per swarm. We present for this metric both the maximum and the mean for the values observed over time for each swarm.

Table 9: Largest swarms and their trackers.

	Largest swarms		
	Top-1%	Top-0.1%	Top-0.01%
Largest swarm trackers			
Number of trackers	94	18	4
Percentage of alive trackers	12.22%	2.34%	0.52%
Max. largest swarms served by a single tracker			
Number of swarms	6,744	946	141
Percentage of large swarms	42.93%	60.21%	89.75%

Giant swarms represent about half of the Top-0.1% of our sample. For the Top-0.01% of the swarms we observe for the maximum number of concurrent sessions a median of 36,080 peers, indicating much higher user participation than in previous measurement studies [15, 25]. Notably, there exist swarms with hundreds of thousands of users.

We focus now on the trackers managing at least one swarm appearing in one of the "top trackers" lists. Table 9 shows information about the top trackers. The number of unique trackers serving all the large swarms is 94 for Top-1%, 18 for Top-0.1%, and only 4 for Top-0.01%. We notice that a small percentage of the trackers manage all of the largest swarms. The largest such tracker manages about half of the swarms in Top-1% and Top-0.1%, and almost all of the swarms in Top-0.01%. Thus, the most popular content is largely served by a single tracker, raising availability concerns.

6. RELATED WORK

We survey in this section related research efforts focusing on BitTorrent and other peer-to-peer file-sharing networks.

BitTorrent There have been many BitTorrent measurement and analysis studies in the past five years. Table 10 characterizes previous studies by size, measurement technique, and special feature. Size-wise, our study provides one of the largest datasets on the status of the BitTorrent network (though we realize that more is not necessarily better). Concerning measurement techniques, our study proposes novel tracker search functionality, including automated search. We also note that BTWorld involves many non-trivial design choices. In contrast, most of the studies reported here use improvised measurement infrastructure, without reporting design or technical details. The remaining two studies, Ios06 [13] and Rod09 [30], develop the MultiProbe and the APOLLO infrastructures, respectively, but these both focus on PirateBay, and therefore have more limited data collection and processing challenges. They also assume that the measurement infrastructure can comprise hundreds of globally-distributed machines, which contrasts to our single-location, few-machines measurement infrastructure. Last, we compare studies by special feature, that is, the feature or aspect of BitTorrent the study was the first to investigate comprehensively. Our study is the first to focus on observing the *global* BitTorrent network; this enables better understanding of the network, including the identification and quantification of two new BitTorrent phenomena, giant swarms and spam trackers.

Two recent studies by Dán and Carlsson [4, 5] are closely related to our work. In particular, they collect information from a large number of BitTorrent trackers (including the largest trackers we investigate), using the "well-known BitTorrent site" and "torrent file parsing" techniques. However, these studies have a different focus from ours, do not use several of the tracker search techniques described in Section 3.3, collect information about fewer swarms (3.3M and 5.2M hashes, respectively, vs our 10.3M), and obtain a majority of their unique hashes from a single BitTorrent community (Mininova).

Other peer-to-peer file-sharing networks Similarly to the case of BitTorrent, much work has been put into measuring peer-to-peer file-sharing networks such as Gnutella [26, 32], FastTrack [27], eDonkey [8], and Overnet [2, 31]. Due to the differences between BitTorrent and these networks, the data collection infrastructures used in these studies face different data collection challenges than ours.

7. CONCLUSION AND ONGOING WORK

As BitTorrent increases in scale and complexity, new measurements are needed for scientific and system optimization purposes. Motivated by the experience of the Internet community, a decade ago, in this work we formulate and take initial steps towards meeting an ambitious challenge: to obtain a global view of the BitTorrent network. We have first presented the design of BTWorld, an architecture for observing the global status of BitTorrent through information provided simultaneously by a large number of popular trackers. Then, we have implemented and deployed BTWorld, periodically collecting information from hundreds of trackers. BTWorld is then used to analyze one week of global BitTorrent operation, which resulted in a large-scale measurement—over 10 million swarms—and a holistic analysis of BitTorrent's performance, scale, and reliability. Last, this broad analysis has enabled us to provide new and more detailed insights about two important BitTorrent phenomena, spam trackers and giant swarms.

We continue to work towards obtaining comprehensive, long-term data about the global BitTorrent status. We are working towards making this dataset available to the research community as part of the Peer-to-Peer Trace Archive (`p2pta.ewi.tudelft.nl`).

We thank the reviewers and especially our shepherd, Dr. Carey Williamson, for their useful comments. This work was partially supported by EC's FP7 programme through projects P2P-Next (#216217) and QLectives (#231200).

8. REFERENCES

[1] N. Andrade, E. Santos-Neto, F. V. Brasileiro, and M. Ripeanu. Resource demand and supply in BitTorrent content-sharing communities. *Computer Networks*, 53(4):515–527, 2009.

[2] R. Bhagwan, S. Savage, and G. M. Voelker. Understanding availability. In *IPTPS*, 2003.

[3] Committee on Research Horizons in Networking, Computer Science and Telecommunications Board, National Research Council. Looking over the fence at networks: A neighbor's view of networking research. Technical report, National Academy Press, 2001.

[4] G. Dán and N. Carlsson. Dynamic swarm management for improved BitTorrent performance. In *IPTPS*, 2009.

[5] G. Dán and N. Carlsson. Power-law revisited: A large scale measurement study of P2P content popularity. In *IPTPS*, 2010.

[6] Ernesto. BitTorrent's Future? DHT, PEX and Magnet Links Explained, 2009. [Online] `torrentfreak.com`.

[7] J. Falkner, M. Piatek, J. P. John, A. Krishnamurthy, and T. Anderson. Profiling a million user DHT. In *IMC*, pages 129–134, 2007.

Table 10: Monitoring studies.

Study	Size (Total)	Technique	Special Feature
Iza04 [15]	1 swarm (Linux distro)	tracker log, custom client	flashcrowd, BitTorrent swarm
Pou05 [25]	SuprNova 1 swarm	web crawl custom client	content management (moderation) BitTorrent site and mirrors
Guo05 [9]	2 trackers, 4.4k swarms, 845k sessions	web scrape, traffic dump	multi-downloads
Ios06 [13]	PirateBay 2,700 swarms, 250k users	web crawl, tracker scrape, custom client, Internet measurement	correlation of peer and Internet characteristics
Neg07 [21]	21k swarms	web crawl, DHT probing	tracker availability
Pia07 [7, 24]	300k users	web crawl, DHT probing	DHT, altruism
And09 [1]	3 sites	web crawl	collaboration
Rod09 [30]	PirateBay, 600 swarms	tracker scrape, custom client, PEX	deviant client detection
Zha09 [34]	9 sites, 35k swarms	web crawl, tracker scrape, custom client	sampling bias
Men09 [18]	66k swarms, 14M users	web scrape, PEX, custom client	content availability
Dán09/10 [4, 5]	721 trackers, 3.3/5.2M swarms	web scrape, tracker scrape	content popularity, load balancing
Meu10 [19]	5 sites, 500k users	web scrape, tracker scrape, custom client	private trackers
This study	**769 trackers** **over 10M swarms**	**tracker search** tracker scrape, **content size search**	**global BitTorrent view** **spam trackers, giant swarms**

[8] F. L. Fessant, S. B. Handurukande, A.-M. Kermarrec, and L. Massoulié. Clustering in peer-to-peer file sharing workloads. In *IPTPS*, 2004.

[9] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, analysis, and modeling of BitTorrent-like systems. In *IMC*, pages 35–48, 2005.

[10] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *AIRWeb*, pages 39–47, 2005.

[11] G. Hazel and A. Norberg. Extension for Peers to Send Metadata Files. Enhancement Proposal 9, BitTorrent, 2008. [Online] `bittorrent.org/beps/bep_0009.html`.

[12] J. Hoffman. Multitracker Metadata Extension. Enhancement Proposal 12, BitTorrent, 2008. [Online] `bittorrent.org/beps/bep_0012.html`.

[13] A. Iosup, P. Garbacki, J. A. Pouwelse, and D. H. J. Epema. Correlating topology and path characteristics of overlay networks and the internet. In *CCGrid Workshops, GP2PC*, page 10, 2006.

[14] Ipoque. Ipoque Internet studies, 2006-2009. [Online] `ipoque.com/resources/internet-studies`.

[15] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. Felber, A. A. Hamra, and L. Garcés-Erice. Dissecting BitTorrent: Five months in a torrent's lifetime. In *PAM*, pages 1–11, 2004.

[16] T. Karagiannis, A. Broido, M. Faloutsos, and K. C. Claffy. Transport layer identification of P2P traffic. In *IMC*, pages 121–134, 2004.

[17] A. Loewenstern. DHT Protocol. Enhancement Proposal 5, BitTorrent, 2008. [Online] `bittorrent.org/beps/bep_0005.html`.

[18] D. Menasche, A. Rocha, B. Li, D. Towsley, and A. Venkataramani. Content availability and bundling in swarming systems. In *Proceedings of CoNEXT*, pages 121–132. ACM, 2009.

[19] M. Meulpolder, L. D'Acunto, M. Capotă, M. Wojciechowski, J. Pouwelse, D. Epema, and H. Sips. Public and private BitTorrent communities: A measurement study. In *IPTPS*, 2010.

[20] A. Miklas, S. Saroiu, A. Wolman, and A. Brown. Bunker: A privacy-oriented platform for network tracing. In *NSDI*, pages 29–42, 2009.

[21] G. Neglia, G. Reina, H. Zhang, D. Towsley, A. Venkataramani, and J. Danaher. Availability in BitTorrent systems. In *INFOCOM*, 2007.

[22] A. Parker. The True Picture of Peer-To-Peer File-Sharing, 2005. Panel Presentation, IEEE Int'l. Workshop on Web Content Caching and Distribution.

[23] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker. A comparison of approaches to large-scale data analysis. In *SIGMOD Conference*, pages 165–178, 2009.

[24] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do incentives build robustness in BitTorrent? In *NSDI*, 2007.

[25] J. A. Pouwelse, P. Garbacki, D. H. J. Epema, and H. J. Sips. The Bittorrent P2P file-sharing system: Measurements and analysis. In *IPTPS*, volume 3640 of *LNCS*, pages 205–216. Springer, 2005.

[26] M. Ripeanu, A. Iamnitchi, and I. T. Foster. Mapping the Gnutella network. *IEEE Internet Computing*, 6(1):50–57, 2002.

[27] S. Saroiu, P. Gummadi, and S. Gribble. Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia Syst.*, 9(2):170–184, 2003.

[28] S. Sen and J. Wang. Analyzing peer-to-peer traffic across large networks. In *Proc. of ACM SIGCOMM IMW*, pages 137–150, 2002.

[29] C. Shannon, D. Moore, K. Keys, M. Fomenkov, B. Huffaker, and K. C. Claffy. The Internet measurement data catalog. *Computer Communication Review*, 35(5):97–100, 2005.

[30] G. Siganos, J. M. Pujol, and P. Rodriguez. Monitoring the Bittorrent monitors: A bird's eye view. In *PAM*, pages 175–184, 2009.

[31] M. Steiner, T. En-Najjary, and E. W. Biersack. Long term study of peer behavior in the Kad DHT. *IEEE/ACM Trans. Netw.*, 17(5):1371–1384, 2009.

[32] D. Stutzbach, R. Rejaie, N. G. Duffield, S. Sen, and W. Willinger. On unbiased sampling for unstructured peer-to-peer networks. *IEEE/ACM Trans. Netw.*, 17(2):377–390, 2009.

[33] J. Yeo, D. Kotz, and T. Henderson. Crawdad: a community resource for archiving wireless data at dartmouth. *Computer Communication Review*, 36(2):21–22, 2006.

[34] B. Zhang, A. Iosup, J. Pouwelse, D. Epema, and H. Sips. On assessing measurement accuracy in BitTorrent peer-to-peer file-sharing networks. Tech. Rep. PDS-2009-005, TU Delft, 2009. [Online] `pds.twi.tudelft.nl/reports/2009/PDS-2009-005.pdf`.