Scientific
Research

A Measurement Study on BitTorrent System

Lin Ye[*], Hongli Zhang, Fei Li, Majing Su

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
E-mail: hityelin@gmail.com

Abstract

Measuring and characterizing peer-to-peer (P2P) file-sharing systems will benefit the optimization and management of P2P systems. Though there are a lot of measurement studies on BitTorrent almost in every important aspect, few of them focus on the measurement issues and the corresponding solutions, which can strongly influence the accuracy of measurement results. This paper analyzes the key difficulties of measuring BitTorrent and presents a measurement system with combination of active and passive ways, which can handle with the problems well and balance the efficiency and integrity. Then compared to other work, a more complete and representative measurement was performed for nearly two months and several characteristics are concerned: 1) there are diverse content sharing in BitTorrent system, but multimedia files that are larger than 100 MB are the most. 2) Distributed Hash Tables has indeed enhanced the ability of peer discovery though there are some pitfalls to be addressed. 3) Pieces are distributed uniformly after the early stage and there are few rare pieces. Furthermore, peer arrival rate shows a periodical pattern, which was not well modeled before. Then an improved model is proposed and the experiment results indicate that new model is fitted in with actual measurement results with high accuracy.

1. Introduction

With the enhancement of PC performance and bandwidth, peer-to-peer (P2P) systems have become immensely popular and attracted millions of users in the past few years. Particularly, BitTorrent has become a ruling heavyweight application that contributes about 53% of P2P traffic. Though BitTorrent scales fairly well and is now widely used in many fields, such as data distribution [1] and media streaming [2], the performance still gets much attention in the literature. With the introduction of Distributed Hash Tables (DHT) [3], BitTorrent has gone from single center to hybrid structure, which might bring about the change of performance.

Nowadays more than fifty kinds of BitTorrent clients have been in use. Unlike other P2P systems such as eMule [4] and Gnutella [5], BitTorrent is simple in design, which is made up of four parts: torrent, peer index, seeds and leechers. A torrent, which is usually uploaded onto a website, is an encoded file that digests the information of sharing files and is necessary for peers to bootstrap themselves into a swarm. Peer index is the set of peers owning the same files. Peer index tracks the status of the peers that are currently active, and acts as a rendezvous point for all peers. So far BitTorrent system has developed three mechanisms for index storage: tracker, DHT and gossip [6]. They are complementary with different working principles. The diversity meets the demand for one peer to connect enough peers to achieve better performance. Peers are divided into two classes according to their states: peers that have already downloaded all files and continue to serve others are called seeds; peers that are still downloading are called leechers.

In BitTorrent, the files are divided into small pieces, and one piece is further divided into smaller blocks. Therefore, a peer can download multiple blocks of the files in parallel, which capitalizes the resources from peers to distribute large contents efficiently. Furthermore, BitTorrent develops a "tit-for-tat" incentive mechanism, which enables peers with high uploading bandwidth to have priority of being served. In this way, peers will pay the penalty for their selfishness, which effectively prevents free-riding behaviors common in P2P systems.

The purpose of this paper is to aid in the understanding of a real and developing P2P system, to provide

measurement data that may be useful in modeling and improving BitTorrent, and to identify design issues in such systems. Our contribution is to discuss the problems and errors in the measurement in depth firstly and design a complete solution to settle them, which is the essential prerequisite to analysis work. We show for the first time the consistency of peers from trackers and DHT, and explore the inherent factors how the differences bring about. Also, we demonstrate the contrast between different torrents in the piece view and propose an improved model of peer arrival rate in the peer view.

The remainder of the paper is organized as follows. Section 2 presents the related work. And then we introduce the measurement system for BitTorrent through analyzing several key difficulties with the corresponding solutions in Section 3. We focus on some new findings in our measurement results in Section 4. Finally an improved model of peer arrival rate is proposed for better fitting in with actual measurement results. We conclude the paper with a discussion of the results in Section 6.

2. Related Work

The amount of P2P traffic and the population of P2P users on the Internet keep increasing. Previous work on BitTorrent has focused on modeling [7,8], incentive mechanisms and improvements [9-11]. In order to get more understanding of BitTorrent, a lot of measurement studies [12-20] have been performed, which can be summarized in four aspects:

1) Torrents: Study [12] collects the information from two popular trackers that continuously update the statistics on their torrents and connected users, revealing that torrents have a wide range in size and the average size exceeds 600MB. Work [13] gathers and parses the web pages of the Supernova, and downloads all torrents. They gives the number of downloads among three types (games, movies and music) and finds that movies are the most popular. Work [14] focuses on the difference between video and non-video swarms. Their results show the torrents shared by video swarms are mostly large while the size of non-video swarms is relatively smaller. But existing measurement on torrents are insufficient due to their limited range with only two trackers or single site, which is sensitive to specific user community.

2) Core components: Work [13] monitors the status of all trackers for a long time and points out the overall performance and stability are strongly influenced by the availability of core components. Study [15] analyzes the prevalence and impact of new mechanisms, including multiple trackers and DHT. There are some findings: a) the introduction of multiple tracker and DHT both can improve availability; b) Trackers might not be inde-

pendent because many of them are hosted in the same machine and multiple trackers may cause swarm splitting; c) Tracker and DHT show complementary characteristic features that trackers provide more information and faster, but DHT can significantly increase the availability of the whole system.

3) Peers: Work [16] depicts the evolution of peers by analyzing five-month tracker log, such as the characteristics of session and geographical analysis. The results demonstrate that BitTorrent is highly effective and can sustain flash-crowd. However, the measurement on single torrent (Linux Redhat 9 distribution) cannot stand for torrents with different popularity. Study [7] finds that the number of peer arrivals decreases exponentially with time in general after its birth. But the work lacks in fine-grained modeling, especially in the peer arrival rate at the early stage. Study [17] reveals that the session length in BitTorrent follows a Weibull distribution more accurately. Work [13] also points out only 17% of peers have an uptime longer than one hour after downloading.

4) Pieces: Studies [18,19] both prove that rarest-first piece selection strategy is better than random strategy. But the simulation results [20] present that the performance benefits provided by network coding in terms of throughput can be more than 2-3 times better in comparison with transmitting unencoded blocks. Different from simulation methodology, study [21] explores the efficiency of rarest-first mechanism by means of instrumented clients that are able to record messages sent or received with the detailed content of the messages, state change in the choke algorithm and other important events. The measurement results show that rarest-first algorithm guarantees close to ideal diversity of the pieces among peers. But limited measurement scope (at most 80 peers) and time (8 hours) cannot give a direct impression on piece distribution in a long term. Work [22] performs a detailed measurement study on the distribution and evolution of piece population. They analyze snapshot data of the near-instantaneous population of pieces, and long-term data of evolution of the piece population over several days. The results validate that the downloading policy of BitTorrent is quite effective in the view of piece distribution and evolution.

Compared with previous work, this paper differs from the following aspects:

- The data are more representative. Our measurement collected about 382,624 torrents from 72 hot websites, which are comprehensive as well as particular only serving cartoon, TV, games and so on. Compared to studies [13,14], the data are more convincing.
- The measurement methods and content are more complete. Several methods including active and

passive ones are integrated to measure BitTorrent system in multilevel view. The combined design can complement each other and solve the limitation of using any single method.

- The difference between torrents is concerned. Two torrents with different popularity are measured continuously for a long time. The details of them are shown in **Table 1**. We can get a deep understanding of different torrents by comparing them.

3. Measurement Methodology

P2P file-sharing systems usually adopt two-level index to maintain the announcement and search of the files, which means descriptive meta data and peer information are separately stored in overlay network. Therefore, a standard process is that users search meta information, choose interested items and then start downloading. If piece information is also viewed as a kind of index, a complete measurement system for P2P should contain three levels:

- Meta measurement: meta data such as filename, file ID, size and type are collected through searching keywords in this level.
- Peer measurement: peer information (IP/Port pairs) is gathered by file ID.
- Piece measurement: in order to get piece information, the system needs to connect as many peers as possible and exchange piece information with each other based on peer measurement.

3.1. Meta Measurement

Most of P2P systems provide an interface to search meta information for users, but BitTorrent has only limited support. The reason lies in the fact that torrents exist on the websites. Users have to browse many pages and choose proper torrents. To gather meta data rapidly and automatically, a crawler based on Nutch is developed and focuses on the following problems:

1) Unrelated Pages Filtering. There are many pages unrelated to torrents on the websites, such as descriptions or advertisements, which will waste a lot of bandwidth to fetch. By analyzing the hierarchical structure of most BitTorrent websites, it is found that URLs on the same site are similar and can be represented in one or more regular expressions. So URLs are clustered to extract the regular expressions and the useful pattern of torrents' links in our work. When crawling the site, we decide whether pages should be fetched according to regular expressions. Furthermore, these filtering rules have the generalization ability and can be adjusted dynamically by the previous result to guide the next fetching.

Table 1. Information of two torrents.

Torrents	Hot	Ordinary
Type	video (movie)	video (cartoon)
Size	457.06 MB	176.15 MB
Website	bbs.wofei.net	bt.ktxp.com
Publishing Time	2009-4-20 16:39	2009-4-20 15:56
Start Time	2009-4-20 16:41	2009-4-20 15:58
End Time	2009-5-19 22:03	2009-5-19 22:05
The Number of Peers	28856	6315

2) Complex Pages Parsing. Complex pages are the ones with a lot of javascripts that involve user interactions to trigger some events for real content, which poses an obstacle to traditional crawlers. Many sites have already used complex pages which make the torrents difficult to obtain. To solve the problem, an ajax parsing engine is designed to deal with the javascripts. The whole process includes parsing javascripts, triggering corresponding events and abstracting real content. The architecture is shown as **Figure 1**.

3.2. Peer Measurement

There are three ways to collect peers: log analysis, passive and active. Although tracker log can record the status of peers more accurately, there are a few drawbacks to be addressed. First, tracker log is not available for anyone, which puts a hard restriction on this method. Second, tracker log does not cover the whole information because more than one tracker may be appointed to a torrent, and it is not practical to obtain their logs all. Third, tracker log cannot represent DHT. Passive method usually deploys several controlled clients in target swarm and waits for incoming connections from others. Passive
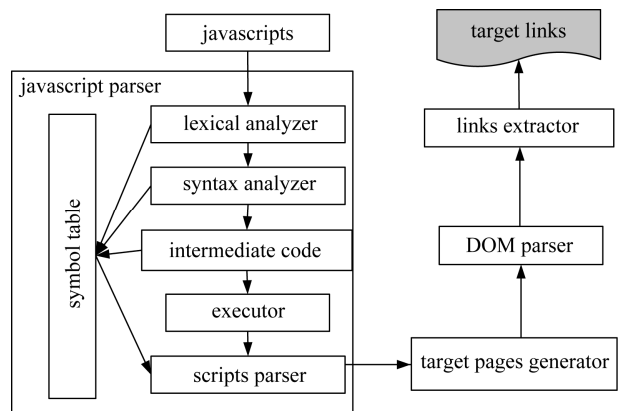

Figure 1. Ajax Parsing Engine.

method is able to handle with the peers behind NAT or firewall, which cannot be connected by external hosts. But passive method needs a long measurement period with low efficiency, especially for a large swarm. Active method acts as a normal user to make requests for other peers to tracker and DHT. However, some mechanisms reduce the efficiency of active method. For example, peer index randomly returns a few peers and does not guarantee the dissimilarity between successive requests. The churn [17] also changes the index at all times, which makes active method hard to converge.

Compared to log analysis and passive method, active method can obtain peers freely and diversely. So this paper mainly implements active measurement on tracker and DHT. To address the limitations, the following settles for bounding or estimating the errors and makes a tradeoff between the efficiency and integrity. To bound the errors, we assume peer index contains N peers for one torrent and responses n peers every request. We denote $T(m)$ as total number of distinct peers that we discovered after m requests and $P(m)$ representing the coverage as the fraction of all peers. Obviously, if $n > N$, $T(1) = N$ and $P(1) = 1$ in one request. If $n < N$, $T(1) = n$ at the first request. Since the index randomly returns peers, we need to avoid the redundancy and $T(m)$ does not increase with linear growth. We suppose $T(m-1)$ is the total number of distinct peers after $m-1$ requests. When the mth request is sent, the probability that undiscovered peers will be returned is:

$$\frac{N - T(m-1)}{N} \qquad (1)$$

The recursion formula of $T(m)$ after m requests is:

$$T(m) = T(m-1) + \frac{N - T(m-1)}{N} \times n \quad m > 1 \qquad (2)$$

The coverage $P(m)$ is:

$$P(m) = \frac{T(m-1)}{N} + \frac{N - T(m-1)}{N} \times \frac{n}{N} \quad m > 1 \qquad (3)$$

According to (2) and (3), for example, more than 230 requests have to be made for measuring a torrent about 5000 peers with 50 peers returned every time, which can obtain 90% of all peers. The process may last 8 minutes or more, and will introduce additional errors into the measurement results. Because every swarm has its own size, there is different measurement time for all of them in real experiments. It is about 10 minutes on average in our measurements for nearly 1,000 torrents. So we disguise ourselves as many legal peers to probe in parallel, which succeeds in speeding up the measurement. To overcome the churn and make the measurement easy to converge, we define $F(m)$ as the number of new discov-

ered peers between successive measurements, which is equal with $T(m) - T(m-1)$. We use a threshold of $F(m)$ to finish the measurement. When $F(m)$ is less than the threshold in successive measurement, we will end up the measurement. From (2), we can conclude that as long as the threshold is less than $n/10$, 90% of peers should be discovered. In order to avoid the influence of randomness, we finish the measurement when the times that the number of new discovered peers is less than the threshold is more than 5, which always empirically discovers 95% of peers.

In addition, when active method is running, many disguised clients that we control are also operating in passive mode, which are able to accept incoming connections and collect the peers that might be behind NAT or firewall. As long as we can own enough peers, the high probability that peers are discovered is guaranteed.

3.3. Piece Measurement

Every peer sharing the same files owns its local piece information, which means a global view of pieces needs the piece information from all peers. In this paper two methods are used as follows:

1) Active: we instrument several clients to connect the peers actively to exchange piece information with each other.

2) Passive: in order to overcome the shortcoming that some peers cannot be connected from the outside. We register many forged users or entities into trackers and DHT, and then wait for incoming connections from others. As long as there are enough controlled forged entities, others will connect them with high probability so as to collect the piece information in a passive way.

Moreover, a traffic monitor system is deployed for passive data collection, which uses deep packet inspection (DPI) to capture the communication between peers and trackers. The parameters of requests from peers to trackers can help us analyze the users and resources, such as what files are sharing and when the peers download and finish. In a word, the proposed system in this paper can carry off all related measurement containing torrents, peers, pieces and user behaviors, whose architecture is shown in **Figure 2**. On the basis of it, a measurement had been performed from 4/2/2009 to 5/27/2009 for about two months.

4. Measurement Results

4.1. Torrents

Size and type are the main static characteristics of the torrents, which can tell what kind of torrents are the most
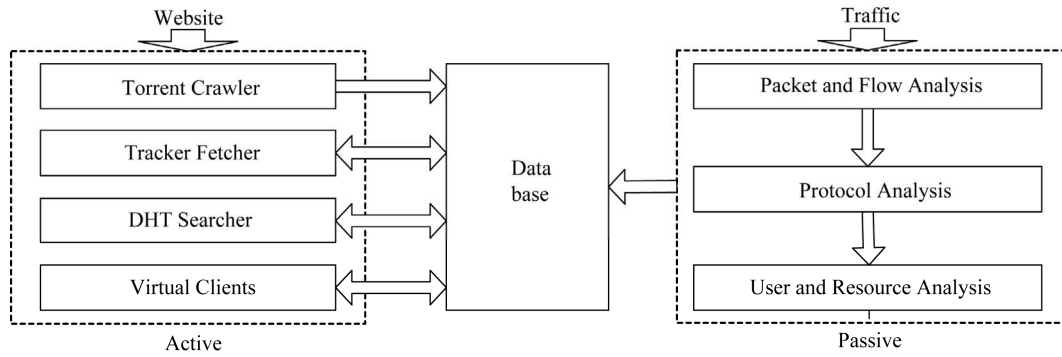
Figure 2. The architecture of BitTorrent measurement system.

popular in BitTorrent. It is helpful to design better mechanisms for resources management and replicas control. **Figure 3** gives the distribution of torrent size with several common bins. About 21% of torrents are more than 1024 MB, which are usually games, HD movies and video collection. 34% of torrents that might be video lie in between 250 MB and 1024 MB. Both of them take up 55% of all torrents, which means large torrents occupy a dominant position in BitTorrent system.

In order to have a close view on real sharing file type, we abstracted the packed filenames from each torrent and counted the corresponding percentage according to file extension. Text files are omitted because they are small and should not be considered as the main content. RAR, MP3, RMVB, JPG and AVI are the top 5 file types with percentage 31.23%, 20.42%, 10.42%, 9.03% and 6.19% respectively. Furthermore, we classified file extension into six types: video, audio, image, executable, archive and other and the corresponding percentage are shown in **Table 2**.

4.2. Core Components

As a rendezvous point, trackers and DHT both play an important role in BitTorrent, which maintain the peers' status and provide other peers for new ones. Study [15] has given a detailed description of tracker type and activity, and drawn a comparison between trackers and DHT in the efficiency of finding peers.

In this paper, we focus on the number of peers and their consistency in trackers and DHT, and explore the inherent factors how the differences bring about. The number of peers is a direct metric presenting the performance of different index. **Figure 4** gives the result of our measurement on ordinary torrent during its first ten days. The total number is the sum of tracker and DHT peers without duplicate ones. Although there are three gaps missing data due to network failure, it has nothing to do with the conclusion we make. First, **Figure 4** shows that with the help of DHT, the total number is almost two

times more than trackers. The diversity of peer index indeed reinforces the availability of BitTorrent. Second, peers from DHT are more than trackers, which implies peers from them may not follow the same pattern. To show the consistency between trackers and DHT, the percentage of the same peers in trackers and DHT is given in **Figure 5**. The high consistency implies DHT makes a good complement to trackers.
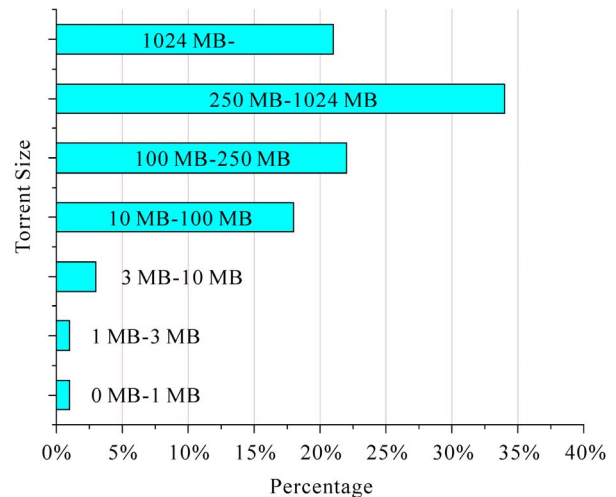

Figure 3. The distribution of torrent size.

Table 2. File type, file extension and corresponding percentage.

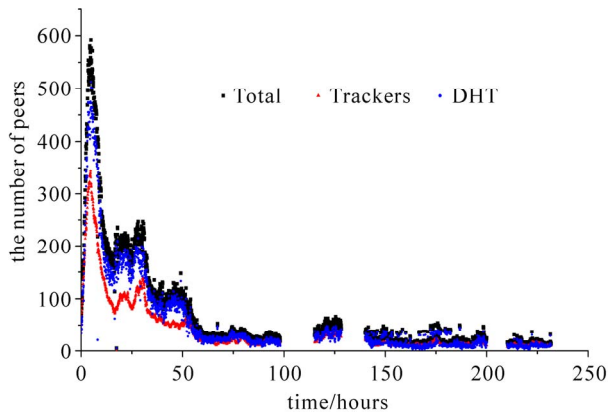| File type	File extension	Percentage
video	rmvb, avi, mkv, mp4, rm, wmv, mpg,…	22.8%
audio	mp3, flac, cbr, wma, ogg, cue, m4a,…	23.5%
image	jpg, png, gif, bmp,…	9.5%
executable	exe,…	0.9%
archive	rar, zip,…	32.1%
other	…	11.2 %

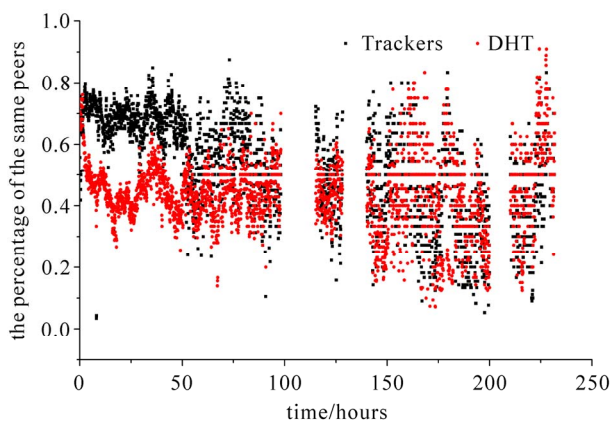**Figure 4. The number of peers in total, trackers and DHT.**

Figure 5. The percentage of the same peers in trackers and DHT.

The number of peers has given a preliminary impression on different index. As above mentioned, we seem to draw a specious conclusion that DHT is better than tracker by the number of peers. Actually, the number of available peers is more convincing than the number of peers. We define a peer is available when the peer can be connected by others. The peers behind NAT or firewall are treated as unavailable ones because they cannot accept incoming requests though they can offer pieces to others in reverse way. The ratio of available peers in tracker on average is 25.3% as the same as the statistics work on peers behind NAT [23], and DHT is 19.1% in **Figure 6**, which shows the ratio of available peers in tracker is higher than DHT.

It is interesting that though there are more peers from DHT than trackers, the peers from DHT are less available. It is the architecture that makes the difference. Tracker is a global component in BitTorrent and can be visible by all peers, but DHT is a completely distributed network. As a result, when some events happen, for example, arrival or departure, tracker can update the peers' status without delay, which keeps the index fresh and
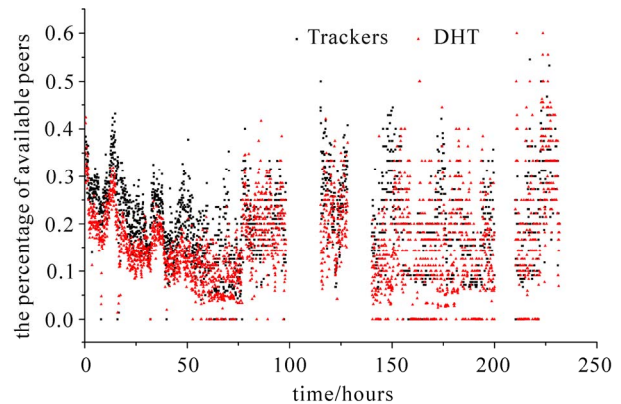

Figure 6. The percentage of available peers in trackers and DHT.

correct. On the other hand, DHT has a complex process to route various messages among peers. A successful event notification needs more than 3-5 messages. Moreover there is no corresponding message for departing, and stale peers are not removed until the pre-set timer is timeout. Many useless peers are left because of delay or forgetting, which wastes a lot of resources and introduces unnecessary traffic.

4.3. Pieces

Pieces are the smallest appreciable unit of data in Bit-Torrent since smaller blocks do not directly affect whether torrent contents finish transferring. The distribution of the pieces across the swarm is important for availability in two aspects. First, if there are not enough replicas for each piece, the whole process will be held due to some missing pieces. Second, if the distribution is not uniform with many rare pieces, the efficiency will be low because the peers owning rare pieces have to afford the pressure from others who want them. Though study [22] was able to obtain 90% of peers by using instrumented clients, it failed to cope with peers behind NAT or firewall. Our combined method can give a more complete view on the distribution. Moreover, we pay close attention to the difference in replicas and rarity of pieces between torrents.

We suppose one file F is divided into n pieces P_1, $P_2, \cdots, P_n$ and there are r_i replicas of piece P_i at time t in the swarm. Therefore, the rate of replicas $R(t)$ is defined at time t:

$$R(t) = \min\left\{r_1(t), r_2(t), \ldots, r_n(t)\right\} \qquad (4)$$

The meaning of $R(t)$ is to give the number of equivalent replicas for the whole file F at time t and reflect the availability of the file in BitTorrent system. The value of $R(t)$ is no less than the number of seeds in existence at

time *t*.

To explore whether the distribution of pieces is uniform, the rate of rarity $C_r(t)$ is defined at time *t*:

$$C_r(t) = \sum_{i=1}^{n} \frac{\left| r_i(t) - r_{avg}(t) \right|}{n} \qquad (5)$$

$r_{avg}(t)$ is the average replicas of all pieces as calculated below:

$$r_{avg}(t) = \sum_{i=1}^{n} \frac{r_i(t)}{n} \qquad (6)$$

The rates of replicas between hot and ordinary torrents are contrasted in **Figure 7**. First, it is shown that at the early stage the rates of replicas are low though the files can be downloaded slowly. Once the seeds leave, the downloading cannot be finished at all. And then the rates go into a steady stage when there are enough equivalent replicas with the best system service ability. The offline of any peer (seed or leecher) has no significant effect on the downloading. With the departure of seeds and the decreasing peer arrival rate, the rates reduce dramatically at the last stage. Second, there is a big difference between two torrents mentioned above. The rate of hot torrent is higher than ordinary one, especially at the last stage, and the steady stage of hot torrent is also longer than ordinary one, which means BitTorrent is more suitable for popular content because it will be difficult for ordinary ones to obtain if users do not find them as soon as possible.

Figure 8 demonstrates the rates of rarity between hot and ordinary torrents, which are similar as a whole. The fluctuation is very drastical at the early stage because the downloading has just begun and only a few pieces have the opportunity to be propagated, and after that the rate gradually approach at 0, which indicates the distribution of pieces is nearly uniform.

5. Model of Peer Arrival Rate

Many studies have already proved the close relationship between the performance and the peer arrival rate in BitTorrent. However, they usually suppose the peer arrival rate follows Poisson distribution, which does not fit in with the results [13]. In fact, in our measurement some new pattern is found in the peer arrival rate, which is also not well modeled before. Before the measurement begins, there are always some peers existing that cannot be considered as new ones in target network. So we use the last result as a point of reference. If some peers are not found in last result but discovered in this experiment, they will be viewed as new ones. **Figure 9** and **Figure 10** show the number of new arrival peers from both torrents.

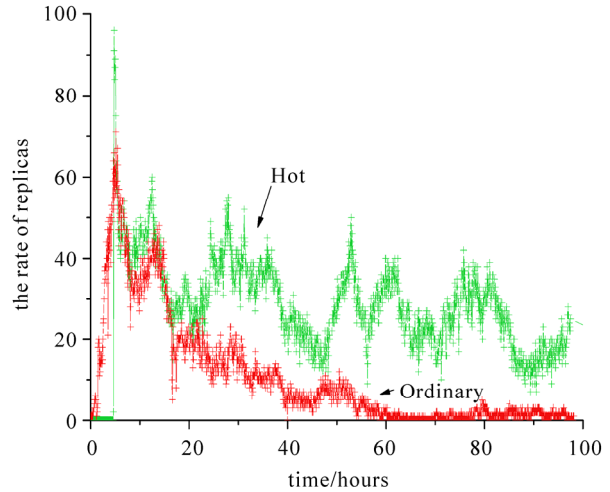From **Figure 9** and **Figure 10**, the number of new arrival
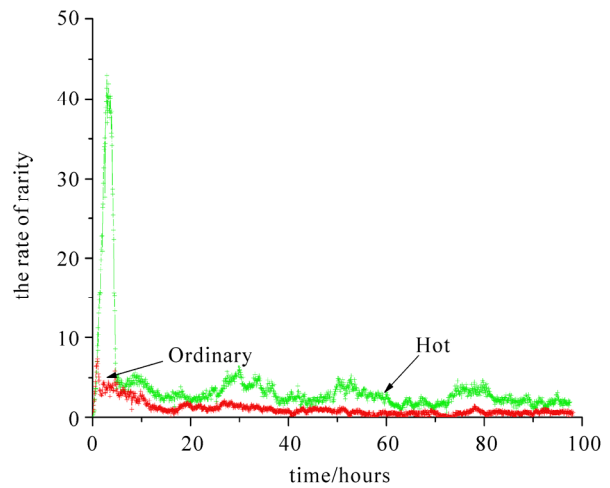

Figure 7. The rate of replicas.

Figure 8. The rate of rarity.

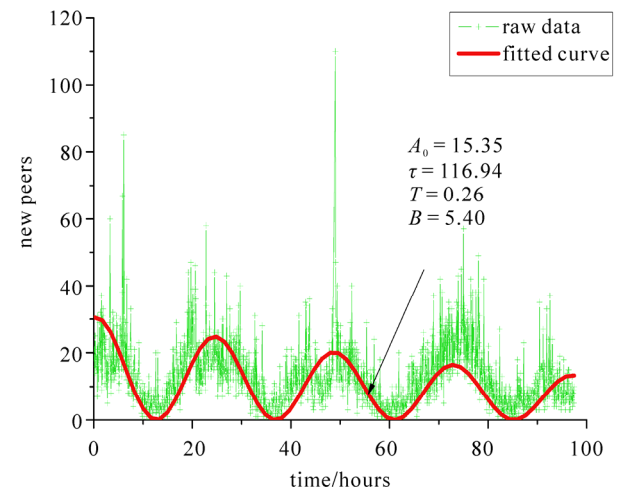$A_0 = 15.35$
$\tau = 116.94$
$T = 0.26$
$B = 5.40$

Figure 9. The number of new peers from hot torrent with fitted curve.
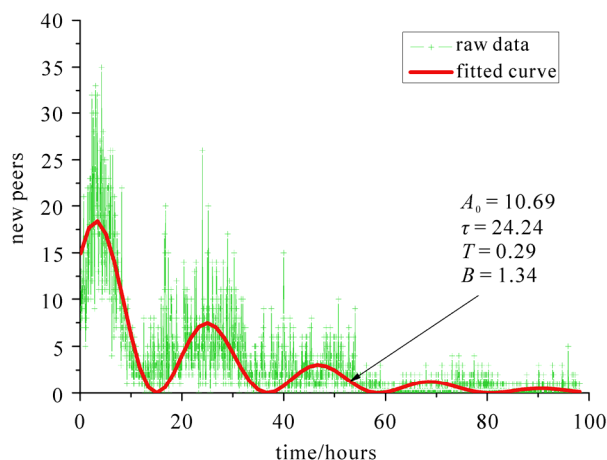

Figure 10. The number of new peers from ordinary torrent with fitted curve.

peers shows a typical fluctuation pattern in both torrents, which seems to follow a daily cycle. Therefore, we count the appearance time of the crest and trough with their corresponding value in detail as shown in **Table 3**. First, **Table 3** can tell that the interval between two adjacent crests or troughs on average is about 24 hours as long as one nature day. This pattern implies the fluctuation seems to follow the behaviors of normal users in the whole day. Second, the value of crests shows an overall decreasing trend with time, although the 3rd crest of hot torrent is an exception.

However, this obvious fluctuation pattern of new arrival peers is not well modeled in the literature as far as we know, so the goal of this paper here is to improve the model of the peer arrival rate in the thought of periodicity. The new model is based on the study [10] as below:

$$\lambda(t) = A_0 e^{-\frac{t}{\tau}}\left(\sin\left(T\left(t+B\right)\right)+1\right) \tag{7}$$

In the above formula, $\lambda(t)$ is the number of new arrival peers at time t. A_0 is the initial oscillation amplitude when the measurement starts, which is related to the popularity of torrents and the state (transient or steady). To be specific, the more popular torrent owns a bigger A_0. τ is the attenuation parameter of the rate. The more popular torrent has a bigger τ, which means slower attenuation. T is the period and B is the phase shift.

In order to quantify the parameters in (7), we define objective function as below:

$$J = \sum_{k=1}^{N}\left(Mod\left(k\right)-Obs\left(k\right)\right)^2 \tag{8}$$

$Mod(k)$ is the kth computation value of the model while $Obs(k)$ is the kth measurement value. This paper uses the BFGS quasi-Newton method to search the para-

Table 3. The Crest and Trough of new arrival peers in both torrents.

Crest/Trough	Hot		Ordinary	
	appearance time (hour)	*value*	*appearance time (hour)*	*value*
Crest 1	6.09	85	4.2	35
Trough 1	12.64	0	14.05	0
Crest 2	22.78	58	24.06	26
Trough 2	36.05	0	40.32	0
Crest 3	49.01	110	50.76	10
Trough 3	60.21	0	62.16	0
Crest 4	75.03	57	73.11	4
Trough 4	84.03	0	86.97	0

meters and make objective function minimum.

Figure 9 and **Figure 10** also give the fitted curves of the periodical model for both torrents. The new proposed model is close to the same with actual measurement results. Furthermore, compared to ordinary one, hot torrent has a higher A_0, which implies hot torrent has a larger network. And hot torrent also has a higher τ, which means the corresponding resource can stay active for a longer time. Considering the value of T in both torrents with separately 0.26 and 0.29, the period is about $2\pi/T \approx 24$ hour.

6. Discussion and Conclusions

The existing measurement works on BitTorrent pay little attention to the measurement issues and the corresponding solutions, which may lead to the inaccuracy of the results. In this paper we design a complete solution to settle them and have presented a detailed measurement study and an analysis of BitTorrent. We believe that this study is a contribution to the ongoing effort to gain insight into a real and developing P2P system.

Though all kinds of torrents are sharing in BitTorrent, the measurement results show that it is more suitable for popular content that are very "heavy". And the high consistency of peers between trackers and DHT implies DHT makes a good complement to trackers. However, DHT still needs considerable improvements for better performance. Also, the fluctuation of peer arrival rate modeled in this paper will cause the difference of system performance, which needs further research, such as incentive mechanisms beyond "tit-for-tat" mechanism and torrents collaboration.

7. Acknowledgements

8. References

[1] B. H. Wei, F. Gilles and C. Franck, "Collaborative Data Distribution with BitTorrent for Computational Desktop Grids," *Proceedings of the 4th International Symposium on Parallel and Distributed Computing*, Lille, 4-6 July 2005, pp. 250-257.

[2] N. Parvez, C. Williamson, M. Anirban and N. Carlsson, "Analysis of Bittorrent-Like Protocols for On-Demand Stored Media Streaming," *Proceedings of the ACM SIG-METRICS International Conference on Measurement and Modeling of Computer Systems*, Annapolis, 2-6 June 2008, pp. 301-312.

[3] F. Jarret, P. Michael, P. J. John, K. Arvind and A. Thomas, "Profiling a Million User DHT," *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, San Diego, 24-26 October 2007, pp. 129-134.

[4] Emule, 2004. http://www.emule-project.net

[5] Gnutella, 2003. http://rfc-gnutella.sourceforge.net

[6] Azureus, 2010. http://azureus.sourceforge.net

[7] L. Guo, S. Chen, X. Zhen, E. Tan, X. Ding and X. Zhang, "Measurements, Analysis and Modeling of BitTorrent-Like Systems," *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, Berkeley, 19-21 October 2005, pp. 35-48.

[8] D. Qiu and R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks," *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Portland, 30 August-3 September 2004, pp. 367-378.

[9] T. Locher, P. Moor, S. Schmid and R. Wattenhofer, "Free Riding in BitTorrent is Cheap," *Proceedings of Hot-Nets-V*, Irvine, 29-30 November 2006.

[10] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy and A. Venkataramari, "Do Incentives Build Robustness in BitTorrent?" *Proceedings of 4th USENIX Symposium on Networked Systems Design & Implementation*, Cambridge, 11-13 April 2007, pp. 1-14.

[11] S. Jun and M. Ahamad, "Incentives in BitTorrent Induce Free Riding," *Proceedings of the ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems*, Philadelphia, 22-26 August 2005, pp. 116-121.

[12] A. Bellissimo, B. N. Levine and P. Shenoy, "Exploring the Use of BitTorrent as the Basis for a Large Trace Repository," Technical Report, University of Massachusetts, Amherst, 2004.

[13] J. A. Pouwelse, P. Garbacki, D. H. J Epema and H. J. Sips, "The Bittorrent P2P File-Sharing System: Measurements and Analysis," *Proceedings of the 54th International Workshop on Peer-to-Peer Systems*, Ithaca, Vol. 3640, 24-25 February 2005, pp. 205-216.

[14] H. Wang, J. Liu and K. Xu, "On the Locality of BitTorrent-Based Video File Swarming," *Proceedings of the 8th USENIX International Conference on Peer-to-Peer Systems*, Boston, 21 April 2009, pp. 1-6.

[15] G. Neglia, G. Reina, H. Zhang, *et al.*, "Availability in BitTorrent Systems," *Proceedings of 26th IEEE International Conference on Computer Communications*, Anchorage, 6-12 May 2007, pp. 2216-2224.

[16] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. A. Felber, A. Al Hamra and L. Garcés-Erice, "Dissecting BitTorrent: Five Months in a Torrent's Lifetime," *Proceedings of 5th International Passive and Active Measurement Workshop*, Antibes Juan-les-Pins, 19-20 April 2004, pp. 1-11.

[17] D. Stutzbach and R. Rejaie, "Understanding Churn in Peer-to-Peer Networks," *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, Rio de Janeiro, 25-27 October 2006, pp. 1-13.

[18] P. A. Felber and E. W. Biersack, "Self-Scaling Networks for Content Distribution," *Proceedings of the International Workshop on Self-Star Properties in Complex Information Systems*, Bertinoro, 31 May-2 June 2004, pp. 1-4.

[19] A. R. Bharambe, C. Herley and V. N. Padmanabhan, "Analyzing and Improving a BitTorrent Networks Performance Mechanisms," *Proceedings of 25th IEEE International Conference on Computer Communications*, Barcelona, 23-29 April 2006, pp. 1-12.

[20] C. Gkantsidis and P. R. Rodriguez, "Network Coding for Large Scale Content Distribution," *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, Vol. 4, 13-17 March 2005, pp. 2235-2245.

[21] A. Legout, G. Urvoy-Keller and P. Michiardi, "Rarest First and Choke Algorithms are Enough," *Proceedings of 6th ACM SIGCOMM Conference on Internet Measurement*, Rio de Janeiro, 25-27 October 2006, pp. 203-216.

[22] C. Dale and J. Liu, "A Measurement Study of Piece Population in BitTorrent," *Proceedings of IEEE Global Telecommunications Conference*, Washington DC, 26-30 November 2007, pp. 405-410.

[23] Free Peers, Inc., "BearShare Network Statistics," 2005. http://www. bearshare.com/stats/