This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Experimental Analysis of Super-Seeding in BitTorrent

Zhijia Chen*, Yang Chen, Chuang Lin*
*Department of Computer Science and Technology,
Department of Electronic Engineering
Tsinghua University, China

Vaibhav Nivargi and Pei Cao
Department of Computer Science,
Stanford University, CA, US

Abstract—**With the popularity of BitTorrent, improving its performance has been an active research area. Super-seeding, a special upload policy for initial seeds, improves the efficiency in producing multiple seeds and reduces the uploading cost of the initial seeders. However, the overall benefit of super seeding remains a question. In this paper, we conduct an experimental study over the performance of super-seeding scheme of BitTornado. We attempt to answer the following questions: whether and how much super-seeding saves uploading cost, whether the download time of all peers is decreased by super-seeding, and in which scenario super-seeding performs worse. With varying seed bandwidth and peer behavior, we analyze the overall download time and upload cost of super seeding scheme during random period tests over 250 widely distributed PlanetLab nodes. The results show that benefits of super-seeding depend highly on the upload bandwidth of the initial seeds and the behavior of individual peers. Our work not only provides reference for the potential adoption of super-seeding in BitTorrent, but also much insights for the balance of enhancing Quality of Experience (QoE) and saving cost for a large-scale BitTorrent-like P2P commercial application.**

***Keywords*-BitTorrent; Super-seeding; download time; upload cost; QoE**

I. Introduction

In BitTorrent-like P2P file sharing application, seeds have a whole copy of the content and many studies have shown that seeds, particularly the initial ones, play a significant role in file distribution performance. How a seed upload pieces to peers, i.e. the seeding scheme, can impact the speed of creating other seeds and the whole downloading Quality of Experience (QoE) of end users. When BitTorrent is adopted by content providers who have to pay for upload bandwidth by the byte, seeding schemes will directly affect the commercial cost. The success of a large-scale commercial BitTorrent application would largely rely on the improvement of peer QoE and the decrease of seed cost.

To meet with those challenges in BitTorrent, Super-seeding, a seeding scheme different from the default one, was first introduced in the BitTornado [4] client in mid 2003, aiming to help the initial seeder with limited bandwidth to "pump up" a large torrent, thus reducing the amount of data it needs to upload to spawn new seeds. Similar feature is also implemented in µTorrent [10] and is called Initial Seeding due to its special application for content initiator.

This work is partly supported by National 863 Program (No.2007AA01Z419), National Science Foundation of China (No. 60702009, No.60473086, No.60473087) and national Basic Research Program of China (No.2007CB310806)

The super-seeding scheme serves as a promising choice to improve the seeding efficiency and decrease seeder uploading cost, but the overall the benefits of super seeding for the whole P2P file distribution remain to be a question. Since the configurations of peers and their upload capacities vary widely, the performance of super-seeding may vary significantly under different circumstances.

In this paper, we conduct an experimental study over the performance of super-seeding, in exploring, whether and how much super-seeding would save uploading cost, whether the overall download time can be decreased by super-seeding, and in which schemes super-seeding works/does not work. Though there have been plenty of work on the performance evaluation of BitTorrent [1, 2 ,6], seldom literature have analyzed the new super-seeding scheme over BitTorrent. Our paper provides an exploring work in studying the effectiveness and application of super seeding of BitTorrent, which may draw much more researches and applications to come. Also, A QoE-targeted and cost-saving view is added during the analysis to meet the needs of a large-scale commercial BitTorrent application. Based on extensive measurements and trace analysis from the experiment in around 250 planet-lab nodes, our study gives out tentative conclusions about the effectiveness of super-seeding and its optimal working circumstances.

II. Mechanism and Effects of Super-Seeding

Inside the super-seeding mechanism, its primary goal is to minimize the cost for seeders to upload. When a seeder, which has initial content to distribute, enters the super-seed mode, it pretends to be a normal client without any data, until some peer connects to it. Then the seeder informs the requesting peer that it has received one piece of data, from which the other peer can now download. When the peer has finished downloading that piece, it is not able to download new pieces from the seed until the seeder finds that the piece it has just sent is present on at least one other peer [4]. In this way, the client does not have access to any other pieces before it distributes what it has received, and therefore does not waste the seed's bandwidth to upload redundant data.

The authors of BitTornado report that a standard seed might have to upload 150% to 200% of the total size of a torrent before other clients become seeds, while the super-seeding seeder may only have to upload 105% of the data [4]. Super-seeding saves uploading cost, but its actual performance in varied circumstances need to be studied.

Another observation is that the super-seeding scheme results in much higher seeding efficiency, by both inducing peers into taking only the rarest content and reducing the amount of redundant data sent. Peers can benefit from super-seeding scheme for creating multiple seeds in a more efficient manner with proper distribution of data pieces. However, this does not mean that the uploading of the whole torrent will take less time. The time to produce the first completion of a downloader is greatly limited by the upload rate of the peers connected to the downloader. Whether super-seeding helps speeded up all peers remain to be measured in real running.

Last, it's clear that, if super-seeding mode is adopted all over the network, it will limit the selection of pieces that a client can download. This will significantly decrease the overall downloading speed, and thus is not recommended. This paper will also identify other circumstances super-seeding does not work.

III. EXPERIMENTAL SCENARIO

To evaluate the performance of super-seeding scheme, we implemented both the regular BitTorrent and super-seeding featured BitTornado version in PlanetLab [5, 6], a world-wide platform for performing Internet measurements. Around 250 PlanetLab nodes with a wide range of sites all over the world were picked and tested in the period from April.2007 to June.2007. Considering the dynamics of Planetlab nodes, the testing time were randomly chosen during those months and the test results were averaged in same time period, i.e day and night. We launched our tracker and seeder in two PlanetLab nodes at Stanford, from which they track and seed other nodes respectively. Same with the settings in BitTornado's specification for super-seeding, the node that functions at super seeding scheme is only the initial content provider. All peers would wait for the unique seeder to provide the content at first. Two files downloading are tested, with a file size of 55.51M and 697.89M respectively. Three mode are measured and compared, i.e, the upload bandwidth of the seeder, is ***above, or equal, or below*** that of the peers.

To meet with the key concerns in large-scale commercial BitTorrent application, we define two metrics for the evaluation, i.e download time and upload cost. The Download Completion Time denotes the time period from when the seeder begins to serve peers to when all peers complete their downloading. Since our concern is to improve the QoE of all peers, we do not adopt the commonly used metric of average download time, but use the overall download time for every peer instead. Considering the dynamics of the network and the heterogeneity of the peers, we deem it more accurate to mark the time when 50%, 90% of the peers finish their downloading.

The second metric measures the seed upload cost: we define Seed Upload Ratio as, $S_{ur} = \dfrac{size\ \ of\ \ total\ \ uploading}{size\ \ of\ \ content\ \ file}$.

Why we vary the upload bandwidth of peers and seeders?

Intuitively, super-seeding changes the behavior of the initial seeder, and thus its effectiveness critically depends on the relative upload bandwidth of that seeder versus that of all other peers.

There are many studies that model the performance of BitTorrent-Like Peer-to-Peer Networks [6-9]. According to the conclusion from the classic fluid model [1], the average downloading time T can be computed as:

$$T = \frac{1}{\theta + \dfrac{1}{\max\{\frac{1}{c}, \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})\}}} \qquad (1)$$

where θ denotes the rate at which downloaders abort the download, c is the average downloading bandwidth of peers. μ is the uploading bandwidth of a given peer. γ is the rate at which seeds leave the system. η indicates the effectiveness of the file sharing.

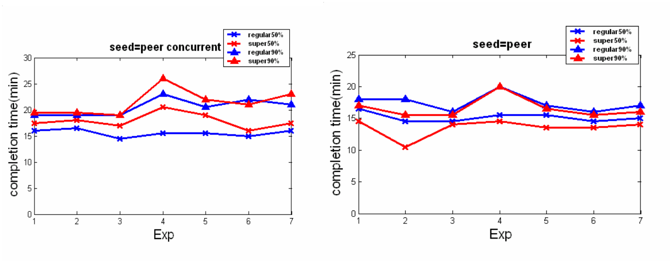
Let's analyze the downloading time in our scenario. Since η can be computed as [1] : $\eta \approx 1 - (\frac{\log N}{N})^k$, which is quite near one. We regard it as the constant parameter one for simplicity here. Then the function $\max\{\frac{1}{c}, \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})\}$ is comes to $\max\{\frac{1}{c}, \frac{1}{\mu} - \frac{1}{\gamma}\}$. In our scenario, the uploading bandwidth μ is set to be smaller than the downloading bandwidth c of most peers (so that we can control the downloading time and observe more details). γ is comparatively much bigger than μ. So the function can be reduced to:

$$T = 1/(\theta + \frac{1}{\frac{1}{\mu} - \frac{1}{\gamma}}) \qquad (2)$$

From above function, we can conclude the key factors that affect the average downloading time. Ruling out the dynamics of PlanetLab peer nodes which affect the value of θ, what influence the system average downloading time most is the value of μ, the uploading bandwidth of a given peer. That is the key reason we vary the relationship of the upload bandwidth of seeders and peers to see the super-seeding's influence over the whole network. γ is also considered in our experimental study when we vary the peer behavior..

Why We don't Run Experiments with Concurrent Seeds

As paper [2] formerly discussed, when we run two independent Bittorrent downloadings at the same machine, there tends to be great difference in downloading time due to the selection of neighbor knowledge. Similarly, we conducted several rounds of experiment to compare the hybrid mode in the same machine by concurrently making the seeder running two independent processes, one in regular seeding, the other in super-seeding, and serving the same other peers respectively.



a) concurrent seeds b) independent seed

Figure 1 : Download completion time in different modes

As fig.1 a) shows, in this concurrent seeds, it seems regular seeding would compete resource with super-seeding, and finally super seeding suffers and takes more time to complete. In comparison, as is shown in fig.1 b), when we implement those two modes independently, the super-seeding scheme can download faster than regular-seeding. We will further justify that this trend is rather stable in this case in later sections. So for the fairness of the experiment and to achieve better performance of super-seeding, we run the two modes independently and avoid concurrent seeds in the same machine.

Why We Use 90% Completion Time as the Metric

As we observed in most of our experiments, around 10% of peers cannot finish downloading in testing time. In selfish mode, this data can go as high as 50% sometimes. If normally a peer can finish downloading within 20 minutes, we would regard it unacceptable to complete in more than 60 minutes, otherwise, the Quality of Experience (QoE) of end users cannot be guaranteed. This is would be a crucial problem if BitTorrent goes to a large-scale commercial application.

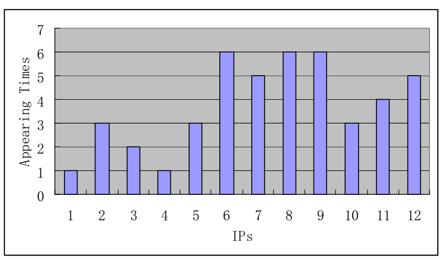


Figure 2. Slow nodes in experiments

To study this issue, we randomly choose one experiment with around 10% unfinished nodes and marked the IP of uncompleted nodes. We then compare those IPs in list of unfinished nodes in another 6 experiments and counted their appearing frequency. As figure 2 shows, most nodes appear more than 3 times in our 7 experiments as unfinished nodes, and some even appear 6 out of 7 times. It should be the network heterogeneity and dynamics that affect their completion. We also notice that some nodes only appeared one time as unfinished nodes, but in all other experiments, completed their downloading rather smoothly. Partly the reason may come from the dynamics of PlanetLab nodes, but possibly it may be a result of the BitTorrent protocol itself. It is unclear whether it can guarantee the downloading for every peer in expected time, either in super-seeding scheme or not. As a solution out, a CDN-assisted BitTorrent scheme may serve as a backup solution to guarantee the QoE of every user. This would a promising area to explore.

IV. EXPERIMENTS VARYING SEED BANDWIDTH

As uploading bandwidth plays a key role in influencing total download completion time, we study the performance of super-seeding in the modes of: upload bandwidth of seed ***above, or equal, or below*** that of the peer. Two key performance metrics are considered: downloading time and seed upload ratio.

A. Seed upload bandwidth == Peer upload bandwidth

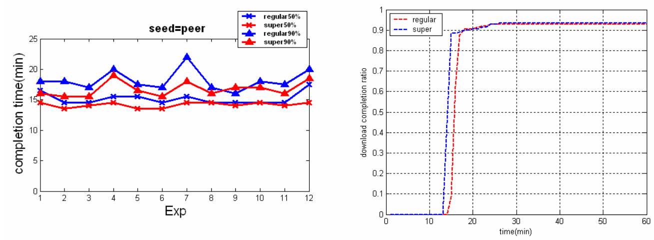
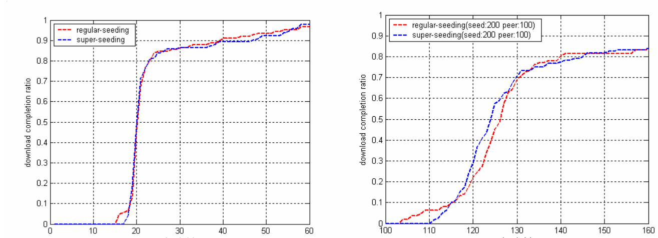


Figure 3. a) Download Completion Time (Seed=peer=75kb/s File: 55.5M); b) Single Experiment Download (seed=peer)

Fig.3 shows 12 sets of experiments, each one of which is the average value of the results of one day during the testing period. With the super-seeding scheme, the whole network would finish downloading faster than the regular mode. To be more specific, we also draw out the whole process of one experiment in fig.3.b), from which we also observe that super-seeding helps speed up overall downloading when the upload bandwidth of seed is equal to that of the peers.

Another interesting observation is that the saving of upload ratio is minimal in this mode with an upload ratio of regular: 124.2% and super-seed: 124.1%. It seems the strength of saving uploading bytes of seeders is not as obvious as that of accelerating downloading when the upload rate seed = peers. We will discuss this in detail in section 2.4.

B. Seed Upload Bandwidth > Peer Upload Bandwidth



a) Seed:100 kb/s Peer:50; file Size: 55.5M; b) S*eed=200 peer=100kb/s file:696.2M*

Figure.4 Single Experiment Download Completion Time(seed>peer)

When upload bandwidth Seed > Peer, our experiments show, roughly, the super-seeding scheme would work better than the regular scheme. But the decease in downloading time is not that obvious as in the mode of upload rate seed=peer.

To get more details, we run two experiments with a file size of 55M and a file size of 700M, expecting a longer downloading time would give some clue on what is going on in the whole process. The interesting finding is that the regular mode tends to produce first few seeds faster than super-seeding, but later on, with more optimized distribution of seeds, the super-seeding scheme catches up and outweighs the regular mode. As is shown in figure 4, in the 55.5 file downloading, the late-started super seeding roughly catches up the regular seeding; with a longer downloading time, super-seeding in the 697.89 file downloading would quickly catches up the early-started regular-seeding, though the left 20% nodes seems to halt in process and unable to finish in our observing time, which is mainly due to the slow nodes as we discussed in section 2.1.

There seems to be a tradeoff between the disadvantages of the super seeding in producing the first few completions of downloading (since super-seeding limits some piece download) and its advantage in the efficiency of producing multiple seeds. Under the mode of upload bandwidth seed>peer, those two influences tend to tussle with each other thus producing little speed increase.

Another observation is that for the regular-seeding, the seed upload ratio is 173.79/55.5=313.1%, but the super-seed upload ratio is 133.6/55.5=240.7%. This is rather obvious and encouraging progress in saving uploading cost. The advantage of the mode seed>peer lies most in this point. We will further discuss this in section 2.4

C. Seed Upload Bandwidth < Peer Upload Bandwidth

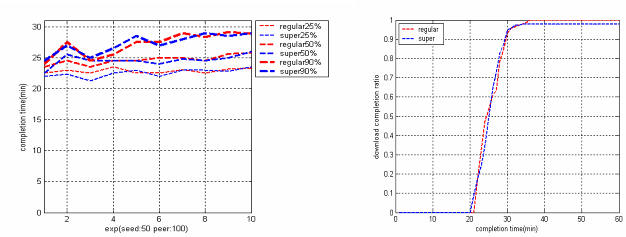


Figure 5. a) Download Completion Time in mode of Seed<Peer
b) Single Experiment Download Time(seed<peer)

As figure 5 shows, over the 10 sets of experiments, the download completion time seems to be almost the same between super-seeding and regular-seeding. During different experiments, the performance of super-seeding varies greatly; sometimes downloads quite faster than regular-seeding but other times performs oppositely. To be more specific, our single experiment shows the download varies in the process and finally achieves minimal difference between super-seeding and regular seeding.

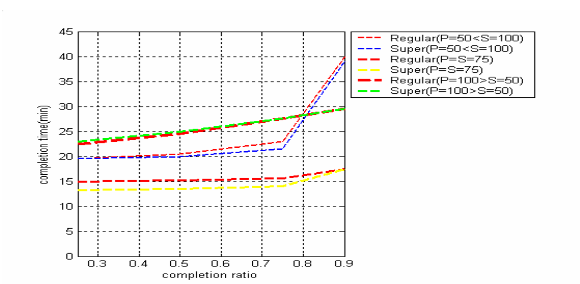
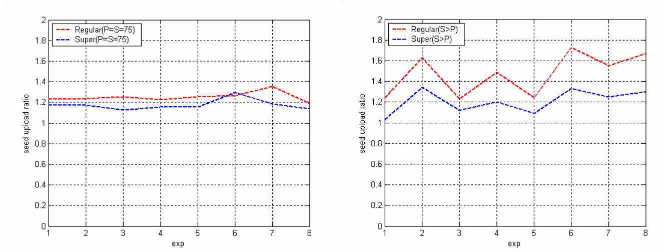


Figure 6. Download completion time under three modes

In summary, as fig.6 shows, for the metric of Download Completion Time, super-seeding tends to perform better than regular-seeding when the upload bandwidth seed is equal to peer. The improvement is rather obvious and stable under this scheme. In scheme of upload bandwidth Seed (S)>Peer (P), super-seeding has some improvements but is not obvious due to its tussle between first few distribution and overall seeding efficiency. In scheme of seed<peer, the difference is minimal and unstable.

D. Seed upload cost under three modes

As figure.7 shows, there is an obvious saving of upload bytes in the mode of upload bandwidth seed>peer. Almost all experiments show that super-seeding scheme would produce over 20% saving of uploading ratio than regular-seeding. In mode of seed=peer, we can also observe the saving ratio but it is not as large as in mode seed >peer.



a) peer=seed b) seed>peer

Figure 7. Seed upload ratio in different modes of upload rate

TABLE I. THREE MODES UPLOAD RATIO

Metric/Mode	R(S>P)	S(S>P)	R(S=P)	S(S=P)	R(S<P)	S(S<P)
90% time	25.7	24.9	18.5	16	26.5	26.6
UP RATIO	229%	180.9%	125.1%	117.5%	121.7%	117.8%

Table 1 gives a summary of those experiments, and we can observe that the saving of upload ratio in the mode of Seed (S)>Peer (P) is most obvious, followed the mode of seed=peer, and the influence over the mode of seed<peer is minimal.

V. EXPERIMENTS VARYING PEER BEHAVIOR

In real P2P network, some peers would function in a selfish manner and leave the network immediately right after they finish the downloading. BitTorrent currently does not have any incentives to encourage users to stay in the system after they have downloaded the file, i.e., when they become seeds. Along with other free-rider behavior, the selfish behavior hinders much of the overall BitTorrent Performance [11]. In this section we will study the influence of selfish behavior of seeds leaving upon the performance of super-seeding.

We design our experiment by making the peers self-kill themselves after they finish downloading, which can well simulate the overall selfish behavior. Considering that the performance of super-seeding is rather stable and obvious under the mode of upload rate seed=peer, we choose it to compare the performance with/without selfish behavior.

Table 2 gives a summary about the download completion time and upload ratio in mode of upload rate seed=peer, with all peers acting in selfish manner. As we formerly discussed, in the mode of upload rate seed=peer, super seeding would obviously help downloading faster and is stable in saving upload ratio. But to our surprise, under the selfish manner, all above conclusion does not hold any more.

TABLE II. SELFISH BEHAVIOR OVER SUPER-SEEDING(SEED=PEER)

TIME/ Mode	R	S	R	S	R	S	R	S
25%	14.5	15.5	14.5	16	15.5	18	16	15.5
50%	15.5	17.5	15.5	19	27	>60	17	24
90%	>60 END 81%	>60 END 87%	>60 END 76%	>60 END 50%	>60 END 50%	>60 END 35%	23 END 90%	》60 END 50%
Up ratio	118.9%	120.0%	119.8%	122.0%	145.9%	133.2%	122.2%	127.4%

R: Regular-seeding **S**: Super-seeding

Under selfish manner, super-seeding may even sometimes download slower than regular-seeding and do not save upload ratio. This shows that selfish behavior harms the advantage of super-seeding.

To be more specific, we draw out the whole process of two experiments in mode of super-seeding (upload bandwidth seed=peer), one in regular manner and one in selfish manner. As figure 8 shows, in regular super-seeding, after the first few downloading completion, new seeds would boom up and most peers would complete the downloading rather quickly. But in selfish manner, the performance would be greatly hindered since the leaving of peers after downloading would decrease the chance of other unfinished peers to complete. We can notice regular super-seeding takes around 1 minute from the first completion to 80% peer completion. But it takes more than 5 minutes for selfish manner to complete 60% after first downloading. After that, the whole network downloading tends to halt due to the negative influence of peers leaving and network dynamics.

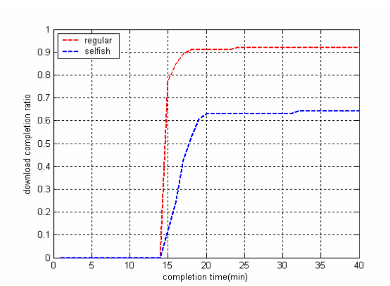


Figure.8 Selfish manner vs regular super-seeding

VI. CONCLUSION

In this paper, we conduct an experimental study over the performance of super-seeding scheme of BitTornado. With the data from our experiments over PlanetLab nodes, we can roughly arrive at the following conclusions:

1) The super-seeding does help saving uploading cost. Though the needed file upload ratio by seed may not always be as low as 105% as BitTornado claims, in most cases, super-seeding can help saving an upload ratio of around 20%.
2) The super-seeding can help deceasing the overall downloading time in certain scenarios. There seems to be a tussle of super seeder's weakness in producing first few completions and its efficiency in making multiple new seeds.
3) The best circumstances to work: if the seed uploading cost is the prior consideration, super-seeding functions best when upload bandwidth seed>peer; if the overall downloading time is the priority, the super-seeding scheme should be adopted when upload bandwidth seed=peer. There is certain tradeoff between the peer download time and seed upload cost.
4) The un-recommended circumstances: When upload rate seed<peer, the advantage of super-seeding is not obvious or stable. Also super-seeding is not recommended for use of every peer, but must strictly limit to the initial content seeder.
5) Factors that hinder the performance of super-seeding: selfish manner, concurrent seeds and network heterogeneity.

Our work provides much insight for further exploring in this area before the adoption of super-seeding scheme in other versions of BitTorrent, and the improvement of Quality of Experience (QoE) for a large-scale BitTorrent commercial application. Due to the dynamics our experiments over PlanetLab nodes and the limit of testing time, we only directly present the observed result and must admit the above conclusion may vary in certain circumstances. For a more stable result and accurate conclusion about the performance of super-seeding, our future work will focus on a large-scale measurement in real Internet and an analytical study over the mechanism inside.

REFERENCES

[1] Qiu, D., and Srikant, R. "Modeling and performance analysis of bittorrent-like peer-to-peer networks". In Proceedings of ACM Sigcomm ,Portland, OR, Aug 2004

[2] Ruchir Bindal and Pei Cao, "Can Self-Organizing P2P File Distribution Provide QoS Guarantees?, Operating Systems Review 40(3): 22-30 (2006)

[3] BitTorrent, www. bittorrent.com

[4] BitTornado, www.bittornado.com/.

[5] S. Banerjee, T. G. Griffin, and M. Pias. "The Interdomain Connectivity of PlanetLab Nodes". In Proceedings of the 5th International Workshop on Passive and Active Network Measurement, 2004.

[6] Guo, L.; Chen, S.; Xiao, Z.; Tan, E.; Ding, X.; Zhang, X.; "A performance study of BitTorrent-like peer-to-peer systems", IEEE Journal on Selected Areas in Communications, Volume 25, Issue 1, Jan. 2007 Page(s):155 – 169.

[7] Rai, Vivek; Sivasubramanian, Swaminathan; Bhulai, Sandjai; Garbacki, Pawel; van Steen, Maarten, "A Multiphased Approach for Modeling and Analysis of the BitTorrent Protocol", 27th International Conference on Distributed Computing Systems(ICDCS '07), 25-27 June 2007

[8] Tian, Y.; Wu, D.; Ng, K. W.; Modeling, "Analysis and Improvement for BitTorrent-Like File Sharing Networks". INFOCOM 2006. April 2006 Page(s):1 – 11

[9] Barbera, Mario; Lombardo, Alfio; Schembra, Giovanni; Tribastone, Mirco; "An Analytical Model of a BitTorrent Peer". 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing(PDP '07) 7-9 Feb. 2007 Page(s):482 – 489

[10] µTorrent www.utorrent.com

[11] Michael Sirivianos, Jong Han Park, Rex Chen and Xiaowei Yang. "Free-riding in BitTorrent Networks with the Large View Exploit". 6th International Workshop on Peer-to-Peer Systems (IPTPS 2007), US