IJCAES
ISSN: 2231-4946
Volume V, Special Issue, January 2015

International Journal of Computer Applications in Engineering Sciences
Special Issue on International Conference on Computing, Communication and Sensor Network (CCSN-2014)
www.caesjournals.org

Impact of Basic Parameters in the Performance of BitTorrent Protocol -A Survey

Abhijit Das, Arup Bhattacharjee
Department of Computer Science and Engineering
National Institute of Technology, Silchar – 788010, INDIA
abhijitcse@yahoo.in, arup@nits.ac.in

***Abstract*-On today's internet, BitTorrent is certainly the most popular P2P file distribution application. BitTorrent introduced some innovative peer and piece selection strategies for efficient file distribution over a large participating community. This popularity motivated researchers in studying the performance of BitTorrent and its mechanisms using a number of approaches. This paper presents a survey of the impact of basic protocol related parameters over performance metrics, including download latency, fairness and service capacity of the BitTorrent protocol. This paper also presents some possible improvements to the existing BitTorrent protocol for improving performance.**

***Keywords* -BitTorrent (BT) protocol, Peer-to-Peer (P2P) networks, Download Latency, Fairness, Service Capacity.**

I. Introduction

Since its inception, peer-to-peer (P2P) paradigm has proved itself to be a scalable, and a robust networking application by providing efficient and effective services for content distribution and personal communication. Unlike the traditional client-server architecture, the essence of P2P lies in combining the resources from all available peers together and distributing the same in return, thus making it (P2P) the most constructive approach for large scale file sharing and distribution. P2P applications first came into the limelight from popular file distribution systems. Today BitTorrent [1] is one of the most popular P2P file distribution systems for large volume of data sharing. BitTorrent (BT) protocol is designed and implemented by Bram Cohen in April and July 2001 respectively [2]. Soon the protocol was incorporated by popular content providers, including Lindows, Blizzard and most Linux distributions. Recent inclusion is social networking giants Facebook and Twitter for efficiently distributing their updates to worldwide servers. BitTorrent provides a scalable way of content delivery, thereby minimizing server congestion, distribution cost and download delay for users.

BitTorrent [1] is a centralized topology file sharing network where a central server coordinates the interaction among peers. But the central server only delivers the directory service, and the distributed peers handle the actual file transfer. Advantages of centralized topology includes, an easier resource management and an efficient indexing and resource discovery. Whenever a peer join or leave the system, the resource index for that peer is added or removed from the central directory of the server. As and when resources are requested, the server looks up its directory and returns the resource location straightway.

A successful implementation of any file distribution application following P2P paradigm demands the support of: (1) a search function for locating the resources the peers are interested in among the participating peers, and (2) a downloading function for downloading the resources from the located peers. BitTorrent protocol however, does not provide search function for getting resource location. Instead, the central server of the topology assists in locating resource by looking up its directory. Additionally, BitTorrent employs some innovative mechanisms, including, tit-for-tat (TFT), optimistic unchoking and rarest first (RF) to facilitate efficient file distribution among the participating peers. The extensive popularity of BitTorrent has appealed several researchers in performing various measures to understand it, and hence among many results identified a significant increase in traffic over the internet. Numerous analysis confirmed that BitTorrent has been generating the maximum P2P file sharing traffic over both the internet as well as LAN. As of February 2013, BitTorrent was responsible for 3.35% of all worldwide bandwidth, more than half of the 6% of total bandwidth dedicated to file sharing [3]. Besides, internet service providers (ISPs) faces substantial traffic-engineering challenges. Major reason being the ignorance of the underlying network topology or ISP link costs in the current implementation of BitTorrent. Consequently, an enormous amount of cross-ISP traffic is generated increasing the operating cost of ISPs [4]. These imperfections provide a scope of further enhancement in the core mechanisms of the BT protocol. For instance, random neighbour selection strategy in BitTorrent application is among the most criticized mechanisms for increasing the download latency and its inefficient usage of network resources. Therefore, some researchers have proposed modified BitTorrent like protocols [5] [6] [7] to enhance performance.

In this paper, we present a survey of the basic BT protocol related parameters & their impact on performance metrics. We summarize the significant findings of these performance studies in tabular form so as to enable us in understanding the behaviour of each performance metric with respect to each protocol related parameter. We also present some suggestion regarding possible improvements in BT protocol. Surveys on BitTorrent performance [8] and modelling [9] are already available in the literature. Presented paper is different from previous studies as this survey explicitly focuses on the impact of basic BT protocol parameters with regard to performance.

The remainder of this paper is organized as follows. In section II, we present an overview of BitTorrent, its mechanisms and working principle. In Section III, we analyse the impact of protocol related parameters over identified performance metrics. We also tabulate our survey with respect to each parameter. Section IV reviews some of the improvement suggestions that we have found during our literature survey for BitTorrent mechanisms to further improve its performance. Finally, we conclude this discussion in section V.

II. Overview of BitTorrent

BitTorrent is one of the most popular P2P protocols for distributing large files. It is implemented in a hybrid P2P system, where peers regularly interact among themselves and occasionally with a server for locating other peers. BitTorrent doesn't download a file from a single source instead, it organizes peers into an overlay network called "Torrent" to distribute a file among themselves simultaneously. A separate torrent needs to be established in order to distribute a file in BitTorrent. We only intend to present a brief description about BitTorrent's working principle. Readers interested in additional information may refer to [1] and [2].

A. Basic Components

1) **Torrent File:** Stores the list of SHA 1 hashes of all the pieces of a file for integrity verification upon download completion.

2) **Tracker:** An active web browser that allows peers to find each other by returning a random list of peers sharing a given content.

3) **Seeder:** A peer that has the complete file that is being shared.

4) **Leecher:** A peer that only has part of the content that is being shared.

5) **Pieces**: File distribution is done by breaking the file into fixed size pieces (typically 256 KB each), and exchanging the pieces among the interested peers.

6) **Chunks:** A piece is further broken into chunks or blocks, typically of 16 KB of size.

B. Protocol

The BT protocol consists of twelve messages: **handshake** (initial message), **bitfield** (indicates the pieces a peer has), **keep-alive** (keeps connections active), **port** (informs on port changes), **choke**, **unchoke**, **interested**, **not interested** (inform the remote peer about state changes), **have** (informs that local peer has a new piece), **request** (requests a piece), **piece** (sends a piece), and **cancel** (cancels a piece request).

C. Operation

There are five basic steps to establish a file sharing process in BitTorrent [8].

1) To download a file, Peer A (any arbitrary peer) first downloads the corresponding .torrent file from a web server.

2) Then, Peer A contacts the tracker for a list of active peers who is participating in the torrent.

3) Next, the tracker returns a list of peers who are involved in the torrent.

4) After that, Peer A adds all the connected peers from the list as its neighbours, and send the file piece request to each other.

5) Finally, once the request is accepted, Peer A can exchange file piece with the neighbours.

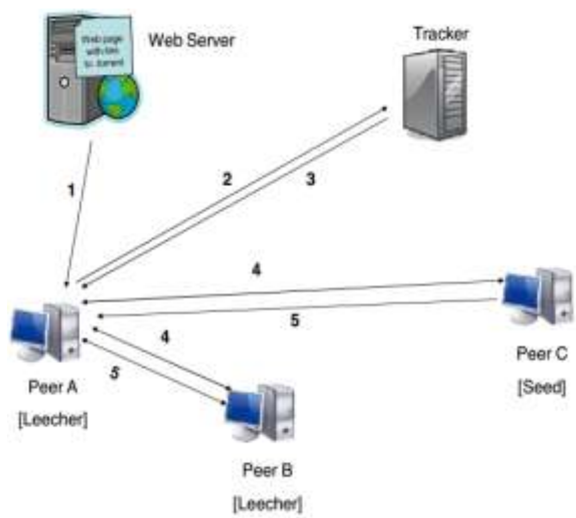

Figure 1: BitTorrent file sharing process [8]

D. Peer Selection Strategy

Different strategies are employed in different scenarios. When together they form the basis of choking algorithm. The motivation behind is to improve the fairness of BitTorrent application.

1) **Tit-for-Tat (TFT):** Each peer uses this mechanism to unchoke four best peers providing highest download rate to it and choke the rest of requesting peers (if any). The TFT mechanism offers an incentive in return to neighbouring peers offering higher download rates, and punishes the free riders. Normally, this mechanism is run every 10 seconds by every peer, and the amount of data received over the latest 20 seconds determines the download rate.

2) **Optimistic Unchoking:** This mechanism is of special importance to newly joined peers to get started. Every 30 seconds, a peer randomly unchoke a fifth peer who sent a request irrespective of its uploading rate. Optimistic Unchoking allows a peer to discover better neighbours to exchange pieces with, as other neighbours may offer higher downloading rate than the currently unchoked peers. Using only TFT mechanism will provide no opportunity to discover peers with higher uploading rate.

3) **Anti-snubbing:** TFT mechanism is executed every 10 seconds to update the unchoked peers. In the process, a peer regards itself being snubbed by another peer if it is choked by the peer it was formerly downloading from. Hence, this peer will choke that peer back as anti-snubbing, thereby not uploading to that peer anymore by regular unchoke.

4) **Upload only:** Seeders have nothing to download so they cannot employ TFT based on higher downloading rates. Instead, this mechanism unchoke peers who serves by utilizing all of their upload capacity [2].

E. Piece Selection Strategy

This is another successful strategy in BitTorrent system, telling the peers to which piece should be downloaded next. An appropriate strategy can increase the file availability over the network.

1) **Strict Priority:** when some chunks (sub-pieces/ blocks) of a particular piece are not received yet, then this mechanism requests for the unavailable chunks of that piece before requesting chunks from any other pieces. Peers will not request another piece until the completion of the current piece as only a full copy of a piece can be traded with others.

2) **Rarest First (RF):** In this mechanism a peer selects the next piece which is the rarest among its neighbours. Here each peer maintains a list containing the number of each piece in its neighbours. This list is updated every time a piece enters or leaves the neighbourhood. A rarest piece set containing the pieces with least copies is determined. Then the rarest piece is selected from the set for download. RF enhances the file availability and diminishes the risk of system death from random seed departure.

3) **Random First Piece:** When a peer first joins a torrent, it cannot employ the RF mechanism. As the peer has no piece at the beginning to trade with others, it tries to quickly download a complete piece to get started with the TFT mechanism. Hence, this peer does not bother about the rarest piece, but any random piece available for download as fast as possible.

4) **Endgame Mode:** Towards the end of a download, a requested piece from a peer having a slow transfer rate can prolong the finishing time. A peer adopts this mode just before the end of its file download by requesting all its neighbours for the chunks it has not received. Once received, the peer cancels the request of this chunk from its neighbours to avoid redundant download hence decreasing the bandwidth wastage.

III. Performance Analysis

BitTorrent is one of the most popular P2P protocols for distributing large files. It is implemented in a hybrid P2P system, where peers regularly interact among themselves and occasionally with a server for locating other peers. BitTorrent doesn't download a file from a single source instead, it organizes peers into an overlay network called "Torrent" to distribute a file among themselves simultaneously. A separate torrent needs to be established in order to distribute a file in BitTorrent. We only intend to present a brief description about BitTorrent's working principle. Readers interested in additional information may refer to [1] and [2].

A. Piece Selection Strategy

P2P network system distributes the task of file sharing among its peers and provides ad hoc cooperation. Analysing the performance of such a system is not straightforward due to numerous reasons, including its unexpected behaviour, heterogeneous bandwidth, free rider and churn. Substantial effort has been put into research on the performance modelling of P2P file sharing system. What follows next are the widely used performance metrics and a brief insight into each one of them [9].

Download Latency: The duration between a file requests being sent out until the completion of downloading that file. It comprises of the query search time and the file transfer time.

Fairness: A well designed system offering scalability demands incentives for contribution. Thus a better uploader should necessarily get a better downloading service as well.

Service Capacity: The overall achievable throughput a system can offer to the downloaders in a torrent. It is a system level measure of resources which takes into account the total effective upload bandwidth from both seeders and leechers.

B. Parameters and their impact on performance

Performance studies of BT protocol mainly includes simulation, mathematical modelling and behaviour measurement on real torrents. But all these techniques one way or another considers some of the basic protocol related parameters to evaluate the performance. In this section, we first identify the most basic parameters considered in the evaluation process over the years. We then analyse the impact of these parameters individually over the identified performance metrics. We summarize the significant findings of these performance studies in tabular form so as to enable us in understanding the behaviour of each performance metric with respect to each parameter.

Number of peers: In BT protocol, a peer plays both the role of a server and a client simultaneously. Each peer contacts other suitable peers to maximize its download rate. Peers with high upload rates will most likely get high download rate. Once finished downloading, a peer might stay online for serving others as a seeder.

Sharma et al. [10] attempted to study the performance of BT protocol and its networking infrastructure through simulation and real client implementation. They investigated the impact of the number of peers on downloading rate. They found that the total download time decreases with increasing number of peers because of increased file pieces availability across multiple peers. However, download time increases beyond a certain limit because of increased network traffic.

Guo et al. [11] have performed extensive trace analysis and modelling to study BT like P2P systems. Their study offers insights into the evolution of single torrent systems and new findings regarding the limitations of BT systems. As the peer arrival rate decreases exponentially in a torrent, the service availability of the corresponding file becomes poor quickly, and eventually it becomes harder to locate and download this file. When the number of peers is small, the random fluctuation of downloading speed cannot be smoothed. They have also shown that the average downloading rate of all torrents is more stable than a single torrent due to the large population.

Yue et al. [12] have developed a general fluid model to study the performance and fairness of BitTorrent like networks. They have found that BitTorrent is not strictly scalable or safe against selfish leechers and incentives for free riders are stronger than previously believed in theory. A session with low utilization of upload bandwidth can also be very efficient with an abundant number of seeders. But the BT protocol lacks incentives to keep seeders around. Consequently, performance can be quite unstable for different sessions with same peer composition. They also found that the combination of optimistic unchoking and unselfish seeder behaviour makes free riding more attractive.

Fan et al. [13] presented the fundamental and delicate trade-off between performance and fairness for BT like protocols through a complete characterization of different rate assignment strategies. They have shown that the current BT protocol is only one particular operating point in the whole design space. They have stated that with the increasing number of peers, the service capacity of the whole system increases as well.

Yang et al. [14] used a fluid model to study the service capacity of P2P file sharing applications under two types of peer arrival pattern, in a flash crowd and in steady state. They concluded that the service capacity scales logarithmically with the number of peers during the flash crowd phase, and in proportion to the number of peers in the steady state.

TABLE I: IMPACT OF NUMBER OF PEERS IN PERFORMANCE

Performance Metric	Impact	Reference
Download Latency	Decreases with increasing number of peers up to a certain limit, then increases. Speed fluctuation cannot be smoothed with less peers.	[10] [11]
Fairness	Current BT protocol lacks incentives to keep seeders around.	[12]
Service Capacity	Increases as the number of peers increases. Scales logarithmically with the number of peers during flash crowd & scales in proportion during steady state.	[13] [14]

Number of pieces: In BT protocol, a file which is being shared is divided into fixed sized pieces of 256KBs, which are further subdivided into blocks (or chunks) of size 16KBs. Thus a block is the basic transmission unit in the protocol. However, a peer can serve a block only after the entire piece is downloaded and verified by the hash function.

Sharma et al. [10] attempted to study the performance of BT protocol and its networking infrastructure through simulation and real client implementation. They have observed that keeping the download rate constant, the download finish time for each peer remains constant irrespective of their download start time. This is due to the fact that peers joining late have more pieces for parallel download.

Yue et al. [12] have developed a general fluid model to study the performance and fairness of BitTorrent like networks. They have found that BitTorrent is not strictly scalable or safe against selfish leechers and incentives for free riders are stronger than previously believed in theory. They have noticed that a distinct relationship exists between file size and seeder population, thus sharing smaller files more efficiently than larger ones. However, sessions of smaller files tend to cease more easily than larger ones.

Ayele [15] in his thesis work described a concept of community ad hoc network and P2P overlay network. The BT protocol establishes a virtual overlay network which addresses peer heterogeneity and dynamics for each local community ad hoc network. BT protocol is implemented in NS-2 simulations for testing its performance according to the defined parameters. They have graphically shown that the download finish time for files with 20KB piece size is lower with the exception that for the farthest nodes the peer showed to have lower finish time for a larger piece size.

Yang et al. [16] presented and validated GPS, a P2P simulator with several desirable properties. Secondly, they have developed a model of BT that is faithful to the protocol specification. They claimed that this is the first BT simulator that models the full behaviour of the protocol. During simulation, they have noticed that the peers joining the downloading process later gradually caught up with the earlier peers. They believed that this was due to the piece selection algorithm, where a late peer has less chance to fail to pick a piece to download, but more chance to find another peer with spare upload bandwidth.

TABLE II: IMPACT OF NUMBER OF PIECES IN PERFORMANCE

Performance Metric	Impact	Reference
Download Latency	Lower for smaller piece size with the exception of farthest node.	[15]
Fairness	Download finish time for each peer remains constant irrespective of the start time.	[10]
Service Capacity	Smaller file sharing is more efficient than larger ones. However, sessions of smaller files tend to cease more easily than larger ones. Peers who join the downloading process later gradually caught up with the earlier peers.	[12] [16]

Potential Set Size: In BT protocol, each peer maintains information on the pieces possessed by its neighbours. A subset of peers among the neighbours that have at least one piece to trade with the peer at a given instance is the potential set of that peer. A peer informs all its neighbours upon receiving a new piece.

Guo et al. [11] have performed extensive trace analysis and modelling to study BT like P2P systems. Their study offers insights into the evolution of single torrent systems and new findings regarding the limitations of BT systems. They have shown that the client downloading performance is affected by the individual behaviour of seeds when the torrent has a relatively small peer population whereas torrents with larger populations have higher and more stable downloading speed.

Rai et al. [17] presented a multi-phased analytical modelling of the BT protocol. Their model suggested that the stability of the BT protocol greatly depends on the number of pieces and the arrival rate of peers. They found that if the initial potential set size is 0, a peer remains with this neighbourhood until new peers arrive. Thus, they suggested that the probability of remaining in the bootstrap phase should be minimum. And this can be accomplished by efficient selection of neighbour set. Thus increasing the potential set size decreases the download finish time.

Al-Hamra et al. [18] conducted extensive simulations to understand the properties of BT overlay. They have focused on average peer set size, the time for a peer to reach its maximum peer set size, and the diameter of the overlay. They have also analysed the impact of overlay properties in efficiency. They have summarized that a larger peer set improves the speed of piece replication. However, this is at the cost of an added load on each peer that has to maintain a larger number of TCP connections. Larger peer set size also increases the efficiency of BT, and a small overlay diameter is a necessary for not sufficient condition for this efficiency.

TABLE III: IMPACT OF POTENTIAL SET SIZE IN PERFORMANCE

Performance Metric	Impact	Reference
Download Latency	Decreases with increasing potential set size. Easily affected by individual behaviour of seeds when the potential set is small.	[17] [11]
Fairness	If the initial potential set size is 0, a peer remains with this neighbourhood until new peers arrive.	[17]
Service Capacity	A large peer set size increases the efficiency of BitTorrent.	[18]

Bandwidth: In BT protocol, the upload bandwidth of a single server is effectively distributed across all downloaders thus reducing the cost for servers hosting large files. The download bandwidth is also divided among multiple sources since a peer simultaneously downloads from multiple peers and not a single central server.

Sharma et al. [10] attempted to study the performance of BT protocol and its networking infrastructure through simulation and real client implementation. They have observed that increasing the upload rate increases the download rate as well, a verification of TFT strategy. And, download finish time improves (reduces) by increasing download rate. However, after a certain upload rate threshold, the download rate decreases, consequently increasing the finish time.

Guo et al. [11] have performed extensive trace analysis and modelling to study BT like P2P systems. Their study offers insights into the evolution of single torrent systems and new findings regarding the limitations of BT systems. They observed that high speed peers finish downloading quickly and then quit the system soon, due to BitTorrent's biased seed service policy in favor of high speed downloaders. This really affects the fairness to peers in downloading, thus an incentive mechanism is needed to encourage seeds to contribute.

Fan et al. [19] proposed a stochastic differential equation method to model the dynamics of BT protocol. They have obtained the analytical expressions of the average number of seeders and leechers, as well as the average file downloading time and the steady state system throughput. They have calculated that having more seeders will reduce the loading time. But when all peers are saturated due to the bandwidth limit, more seeders bring no improvement in the performance.

TABLE III: IMPACT OF BANDWIDTH IN PERFORMANCE

Performance Metric	Impact	Reference
Download Latency	Reduces by increasing download rate but after a threshold finish time increases.	[10]
Fairness	Peers with high speed finish downloading quickly and then quit the system soon, which defeats the design purpose of the seed service policy.	[11]
Service Capacity	When all peers are saturated due to the bandwidth limit, having more seeders will not improve the performance of end users.	[19]

Number of Connections: In BT protocol, the number of active connections at a peer changes with a general birth/death process. Success rate of new encounters determines the birth rate whereas, the expected length of active connections determines the death rate. A high birth rate and a low death rate ensure high download efficiency.

Sharma et al. [10] attempted to study the performance of BT protocol and its networking infrastructure through simulation and real client implementation. They investigated the impact of the number of peers on downloading rate. They have concluded that although, number of connected peers does help in download rate considerably due to sharing of file pieces, however, beyond a point, it results in a decrease in download rate (speed).

Fan et al. [19] proposed a stochastic differential equation method to model the dynamics of BT protocol. They have obtained the analytical expressions of the average number of seeders and leechers, as well as the average file downloading time and the steady state system throughput. They have observed that a large number of connections will burden the peers with too many connection overheads and will eventually lead to bandwidth saturation.

Cuevas et al. [20] presented a comprehensive study of the overlay topology structure and the connectivity properties at peer level of real BitTorrent swarms. For this purpose they leveraged information collected from 250 real torrents and more than 150k peers. Their results reveal that a significant fraction of peers presents a clear locality-biased composition of both their neighbourhoods and their set of stable neighbours. They have also observed that a leecher keeps just a handful of stable connections with its neighbours with which it exchange most of its traffic. Conversely, seeders do not keep long-term connections with other peers in order to guarantee the homogeneous distribution of pieces among the participants.

TABLE III: IMPACT OF NUMBER OF CONNECTIONS IN PERFORMANCE

Performance Metric	Impact	Reference
Download Latency	Beyond a point, increasing connections results in decrease in download rate.	[10]
Fairness	A significant fraction of peers present a clear locality-biased composition of both their neighbourhoods and their set of stable neighbours.	[20]
Service Capacity	Large number of connections saturate the peer bandwidth.	[19]

IV. SUGGESTED IMPROVEMENTS

The popularity and success of BitTorrent has attracted several researchers to conduct various performance analysis. Subsequently, a number of improvements have been suggested to the existing peer and piece selection mechanisms. In this section, we will mainly present the suggested improvements we have encountered while surveying our performance metrics.

Guo et al. [11] have proposed an architecture for inter-torrent collaboration under an exchange based instant incentive mechanism to address the problem of poor service availability, fluctuating downloading performance, and unfair services to peers. Preliminary simulations have shown promising results.

Fan et al. [13] proposed a simple design knob which can be implemented in a distributed manner. They have quantified the performance merits, both in average downloading time and fairness, as they varied the design knob. Performance evaluation has been carried out to quantify the merits and properties of these BT-like protocols.

Yang et al. [14] have studied that Multi-part combined with parallel uploading when properly optimized will generally improve system performance, particularly when peers exit the system at a high rate. A credit system might help provide peers incentives to share and thus improve performance.

Hamra et al. [18] have identified that the maximum peer set size is a trade-off between efficiency and peers overhead, and they have also explained why the maximum number of outgoing connections must be set to half of the maximum peer set size.

V. CONCLUSION

This paper presented a survey on the impact of basic BT protocol related parameters on different performance metrics. We identified the most widely considered performance metrics, Download latency, fairness and service capacity and observed the impact of most basic parameters including number of peers, number of file pieces, potential set size, bandwidth and number of connections on them. We tabulated our significant findings with respect to each parameter for better understanding and realization. We also presented improvement suggestions to the existing BT protocol that we have encountered during the survey of performance metrics. Thus we conclude this study of the basic parameters and their impact on the identified metrics with a motive to come up with a better design and performance modelling of the BT protocol in near future.

REFERENCES

[1] BitTorrent, http://www.bittorrent.com/.

[2] B. Cohen, Incentives build robustness in BitTorrent, First Workshop on Economics of Peer-to-peer Systems, USA, 2003.

[3] Application Usage & Threat Report, Palo Alto Networks, 2013, Archived from the original on 31 October 2013, Retrieved 7 April 2013.

[4] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, Improving traffic locality in BitTorrent via biased neighbor selection, 26th IEEE ICDCS, pp. 66, 2006.

[5] C. Gkantsidis and P. Rodriguez, Network coding for large scale content distribution, IEEE INFOCOM, vol. 4, pp. 2235-2245, 2005.

[6] D. Levin, R. Sherwood, and B. Bhattacharjee, Fair file swarming with FOX, 5th IPTPS, 2006.

[7] R. Sherwood, R. Braud, and B. Bhattacharjee, Slurpie: a cooperative bulk data transfer protocol, IEEE INFOCOM, vol. 2, pp. 941-951, 2004.

[8] R. L. Xia, and J. K. Muppala, A survey of BitTorrent performance, IEEE Communications Surveys & Tutorials, vol. 12, no. 2, pp. 140-158, 2010.

[9] K. Xu, Performance Modelling of BitTorrent Peer-to-Peer File Sharing Networks, Cornell University Library, 2013.

[10] P. Sharma, A. Bhakuni, and R. Kaushal Performance Analysis of BitTorrent Protocol, IEEE NCC, pp. 1-5, 2013.

[11] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, A Performance Study of BitTorrent-like Peer-to-peer Systems, IEEE Journal on Selected Areas in Communications, vol. 25, no. 1, pp. 155-69, 2007.

[12] Y. Yue, C. Lin, and Z. Tan, Analyzing the Performance and Fairness of BitTorrent-like Networks Using a General Fluid Model, Elsevier Computer Communications, vol. 29, no. 18, pp. 3946-956, 2006.

[13] B. Fan, J. C. S. Lui, and D. M. Chiu, The Design Trade-Offs of BitTorrent-Like File Sharing Protocols, IEEE/ACM Transactions on Networking, vol. 17, no. 2, pp. 365-376, 2009.

[14] X. Yang, and G. D. Veciana, Service Capacity of Peer to Peer Networks, IEEE INFOCOM, vol. 4, pp. 2242-2252, 2004.

[15] E. D. Ayele, Analysis and deployment of the BitTorrent protocol for Community Adhoc Networks, M.Sc. Thesis, Delft University of Technology, 2011.

[16] W. Yang, and N. A. Ghazaleh, GPS: a general peer-to-peer simulator and its use for modeling BitTorrent, 13th IEEE MASCOTS, pp. 425-432, 2005.

[17] V. Rai, S. Sivasubramanian, S. Bhulai, P. Garbacki, and M. V. Steen, A Multiphased Approach for Modeling and Analysis of the BitTorrent Protocol, 27th IEEE ICDCS, pp. 10, 2007.

[18] A. A. Hamra, A. Legout, and C. Barakat, Understanding the Properties of the BitTorrent Overlay, Cornell University Library, 2007.

[19] B. Fan, D. M. Chiu, and J. C. S. Lui, "Stochastic Differential Equation Approach to Model BitTorrent-like P2P Systems," IEEE ICC, vol. 2, pp. 915-920, 2006.

[20] R. Cuevas, M. Kryczka, A. Cuevas, C. Guerrero, and A. Azcorra, Connectivity properties of real BitTorrent Swarms, KSII Transactions on Internet and Information Systems, vol. 7, no. 9, pp. 2246-2267, 2013.