Journal of Theoretical and Applied Information Technology
10th November 2013. Vol. 57 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# METHODOLOGIES TO IMPROVE THE PERFORMANCE OF A BITTORRENT LIKE PEER TO PEER NETWORK

**[1] M.PADMAVATHI , [2] R.M.SURESH**

[1] Associate Professor, Department of Information Technology,
RMD Engineering College, PO box 601 206, Kavaraipettai, Chennai ,
[2] Principal, SMIT Engineering College, PO box 600069,
Near Mangadu, Chikkarayapuram, Chennai
Email : [1] mpadmavathi1979@gmail.com

**ABSTRACT**

In a Peer-to-Peer (P2P) network there is a lack of clear distinction among client and server nodes. Each and every network node (or peer) requires network resources and contributes resources to the network. This arrangement gives a clear representation that when more number of peers enter in to the network, additional resources such as CPU, storage, and bandwidth will be available for all the peers in the network. BitTorrent is a most popular and predominant peer to peer file distribution mechanism due to its scalability and ability to distribute large files quickly and expeditiously without devastating the server capacity. The Study of bittorrrent performance helps in understanding the characteristics and improvements in certain areas of the BitTorrent. This paper proposes new strategy to study and evaluate BitTorrent and its core mechanisms , the performance factor including uplink/ downlink utilization and fairness

**Keywords** : *Peer-to-Peer network, BitTorrent, Uplink Utilization , Downlink Utilization , Fairness*

## 1. INTRODUCTION

Peer-to-Peer (P2P) applications are most widely used internet networking utilities. P2P's inborn characteristics ensure its advantages in content distribution when compared to client-server architecture. In a peer to peer networks there is no clear seperation between client and server nodes as a peer can act as both client and server simultaneously [2]. Time needed for data dissemination is less when compared to client-server methods. Another advantage of peer to peer network is scalability as it can accommodate n number of nodes. BitTorrent seems to be a most popular P2P content distribution system accounting for 50 % to 70% internet traffic and delivers huge volumes of data from an initial server to client sets, but it differs from traditional peer to peer application as it is peer dependent for sharing the content [2]. It caters to large scale Internet use through a unique content distribution mechanism. Figure 1.1 illustrates the block diagram of a peer to peer network.

*Fig. 1 : A Peer to Peer network*

BitTorrent [1] is an emerging very popular and scalable peer to peer content distribution tool. In a BitTorrent, a file is splitted into a large mass of blocks and peers on downloading their first block will start serving other peers. Peers will prefer to download the blocks that are most rare among their local peers in order to maximize their utility to other peers. These approaches allow BitTorrent to use the bandwidth among the peers most effectively in order to handle the flash crowds in a proper manner. In addition to it, BitTorrent also includes a tit-for-tat (TFT)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

incentive mechanism whereby nodes preferentially upload to peers from whom they are able to download at a faster rate. This mechanism is particularly important since studies have shown that many nodes in P2P systems prefer to download the content without sharing anything [15]. The studies indicate that the BitTorrent has accounted for a large share of P2P Internet traffic. The analytical measurement and studies [8, 11, 12] illustrates that the BitTorrent can scale well and can handle large volume distributions more efficiently. However, these studies unanswers several issues like last block problem of LRF Policy , link utilization impaired by random policy and bandwidth policy, fairness criteria of TFT policy , about stability and scalability of the system , Performance of the system in terms of bandwidth utilization. The answers depend on various parameters that a BitTorrent uses. It is very tricky to answer these questions in a live measurement or an analytical setting. Hence, it is better to answer these questions using a simulator based approach which models the data-plane of BitTorrent. The details about the simulator settings and experimental results is described in Sections 4 and 5. The principal findings about the Bittorrent are : Firstly, It scales well in any environment due to the increase in number of nodes and gives an assurance in terms of high uplink bandwidth utilization. Next, the new peers who have no packets initially become the active member of the network by contributing more. However it is possible during a flash crowd situation. This paper gives a performance study of block picking policy like LRF , Random , Bandwidth based policy) in terms of link utilization.

The rest of the paper is organized as follows: Section 2 gives a brief idea about the BitTorrent file sharing process and its mechanisms. Section 3 discusses literature review. Section 4 describes the simulator details. Section 5 presents simulation results under a variety of workloads. Finally, Section 6 provides conclusions and future enhancements.

## 2 BITTORRENT

This Section provides some basic terminologies used by the the BitTorrent, overview of the BitTorrent protocol and its file sharing process, inherent Bittorrrent mechanisms. A complete description of the BitTorrent protocols and its mechanisms are provided in [4].

### 2.1 Basic Bittorrent Notations

**Peer :** A peer can be either a Seeder or a leecher . Seeder is a peer that has a complete copy of a file and involved in an upload process. Leecher is a peer that is still downloading and has a partial copy of the file.

**Pieces and sub pieces** – Piece is the basic nuclear component of a file. Files to be shared are divided into an equal sized pieces tin order to facilitate parallel download and upload. The standard size of the piece is 256 KB [4, 17]. The pieces are distributed randomly among the peers. Each piece is divided into a sub piece called as chunks or blocks. The typical chunk size is 16 KB [4, 16].

**Initial seeder:** It is a seeder who builds the torrent file and publishes it for the sake of other peers to download.

**Peer set:** It is the set of active neighbors for a particular Bittorrent peer. The data conversion is allowed only among a peer set.

**Peer Swarm:** Swarm is a place where a group of seeders and leechers can participates in a file exchange.

**Web Portal:** It is a server on which the content publishers can upload .torrent files and those .torrent files can be downloaded by a bittorrent client.

**.torrent file :** This file contains information about the files to be shared in a network including content name, file size, number and size of the pieces that form the content ,SHA-1 hash values and IP address of the tracker managing the swarm associated to the file.

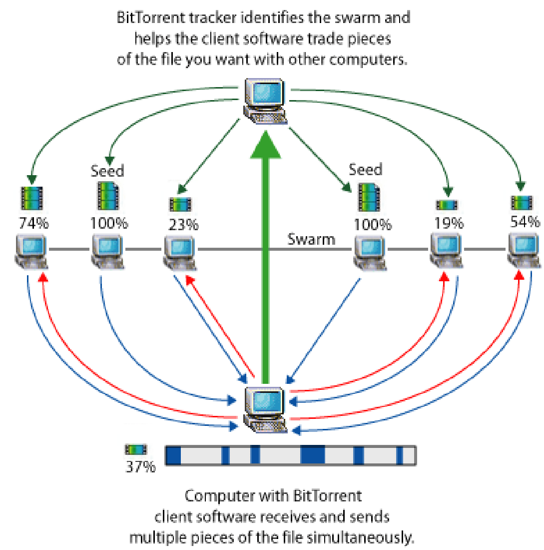
**Tracker:** It is the central component that coordinates a peer swarm and keeps a count of the number of active peers within a group. The tracker does not involve in the content transfer but keeps the statistics of all the peers currently participating in the torrent. The central goal of the tracker is that it does not allow any unauthorized peer to communicate with server and ensures security.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

**Peer State :** A Peer can be in two states namely choked and interested. In a *Choked* state uploader does not want to send anything on his communication link. In an *interested* state the other end has some pieces that the downloader is interested in.

### 2.2 Bittorrent Model

BitTorrent [3] is a predominant peer to peer file sharing protocol facilitating a huge file distribution by means of leveraging upload bandwidth of downloading peers. The idea of BitTorrent file sharing is to split a file into an equal sized blocks allowing nodes to join in a group for parallel uploading and downloading. The file to be downloaded is named as a 'torrent', and can vary from 32 to 256 KB. A swarm is a place where a group of peers can involved in downloading and uploading. The ability of a BitTorrent to upload a torrent blocks before downloading ensures its efficiency among the other peer to peer protocols. Camouflaging of request-response latency is achieved by means of dividing blocks into sub-blocks there by enabling pipelining request [4]. Tracker is a distributed central component which records the status of each and every peer, joining or leaving a torrent. Peers are either seeders or leechers; seeder is a peer with a complete copy of a torrent file and is ready for uploading it to others in order to make others as seeds. Peers which still downloads the torrent is called a leecher. Whenever a peer joins a torrent it obtains a random list of seeders and leechers from the tracker. The peer contacts around 40 nodes as its neighbors. If this number falls below the threshold value say 20 then a fresh list is got from the tracker in order to add a neighbor peers. When a peer gets connected with neighbor peer, it can download and upload blocks with them. Once a peer completes the download process it becomes a seed allowing a spreading of file like a flood among the peers. As more number of peers join the swarm, the probability of a successful download increases. With reduced hardware, The advantages include bandwidth cost, and reduced dependence on the source server and redundancy against system problems. BitTorrent system employs local rarest first (LRF) policy in order to download the blocks from neighbors [5]. LRF chooses the block that is replicated in a least quantity among the neighbors.To ensure data trading, a tit-for-tat (TFT) policy is employed which allows nodes to upload to neighbors which provides the best download rate [6]. Each node is limited by a concurrent upload of about 5 peers and is done by a mechanism called choking, Usually, concurrent upload is limited to 5 ie. at a particular point only 5 chosen neighbors are unchoked. A node reevaluates the download rates received from the neighbors periodically to determine whether a currently unchoked neighbor to be choked and replaced with some other.




Simulation-based approaches facilitates the understanding and deconstructing the BitTorrent performance. Live torrent participation , Live internet measurements like tracker logs [7], alone are not enough for studying the performance metrics. Also BitTorrent configuration mechanisms or an alternate methods can not be incorporated. Simulator used here is an octosim simulator which models and studies the peer activity like joining time or leaving time of a torrent, block sharing etc. All the mechanisms including LRF, TFT are discussed in detail.. Simulations are carried out based on active data and the scenarios are modeled theoretically leading to a better understanding of peer to peer networks.

### 2.3 Bittorrent –like Systems :

**2.3.1 Slurpie :** It is a peer-to-peer protocol meant for bulk data transfer. Slurpie is designed especially to reduce the client download times and to reduce load on servers It employs a new adaptive downloading strategy in order to increase the client performance, and provides a randomized backoff strategy to control the load on the server preciously.

**2.3.2 Grid Torrent :** It is a modified BitTorrent protocol, equipped with modern Grid middleware components. Grid-Torrent is used

Journal of Theoretical and Applied Information Technology
10th November 2013. Vol. 57 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

specifically to transfer the files directly from established GridFTPservers or other GridTorrent peers that are requesting the similar information.

**2.3.3 Avalanche :** It is similar to BitTorrent, but it uses network coding technique to make the content propagation efficient and easier. Like BitTorrent, nodes join the system by contacting a centralized server which provides a random subset of other users. Nodes also employ rarest first algorithm to decide which block to transfer.

**2.3.4 NetStitcher :** It is a system for connecting together the unutilized bandwidth across the different data centers, and using it to carry bulk traffic for backup, replication, or data migration applications. It collects the information about left over resources and uses a store-and forward algorithm to schedule data transfers and adapts according to the resource fluctuations.

**2.3.5 GatorShare :** It is a data management framework that offers a file system interface and an extensible architecture designed to support multiple data transfer protocols, based on which we implement a cooperative data distribution service for Distributed Grids. It eases the integration with Desktop Grids and enables high-throughput data management for an unmodified data-intensive applications.

**2.3.6 AntFarm :** It is a content distribution system based on managed swarms. Antfarm achieves high throughput by considering content distribution as a global optimization problem, where it aims to minimize the download latencies for the participants depending on bandwidth constraints and swarm dynamics.

**2.3.7 BitHoc :** It is an enhancement over BitTorrent, which aims to minimize the downloading time of the content and at the same time enforces cooperation and fairness among peers. It also aims at improving the sharing ratio and the reusability of network resources by means of creating the diversity of pieces in the network.

**2.3.8 FOX :** It is called as Fair Optimal exchange protocol to achieve a fair file swarming. FOX provides an effective optimal downloading time for peers.

### 2.4 Methods to Improve Performance

#### 2.4.1 Chunk Selection Strategy

Whenever a peer wants to download some pieces from its neighbors, it employs various piece selection strategies, which includes the following four policies :

**2.4.1.1 Strict Priority Policy**

In BitTorrent, peers concentrate on downloading a whole piece before requesting another piece. Thus if a sub-piece is requested, then subsequent subpieces of the same piece will be requested preferentially in order to complete the download of the whole piece as soon as possible, because only complete pieces can be traded with others.

**2.4.1.2 Rarest First Policy :**

It follows the principle that it downloads the pieces that are very rare among their neighbor peers. The algorithm works as follows. Every peer keeps a record of a list of pieces that its neighbors possess. The list is revised when a copy of a piece is available from its neighbors. It then bulids a rarest-piece set which is the list of pieces that have the least replicated copies among its neighbors. To download, it then selects the piece that is rare among its neighbors. This strategy ensures that all the pieces are distributed quickly to at least some of the leechers [2] and also reduces the server burden.

**2.4.1.3 Random First Policy :**

In this policy , the first pieces are chosen randomly in order to get a complete piece as quick as possible in order to get ready to respond for the TFT algorithm.

**2.4.1.4 Endgame Mode Policy :**

This mode is adopted by a peer at the end of downloading the file. If a piece with a slow transfer rate is requested from a peer the downloading time will be prolonged. To solve this issue , a peer requests its neighbors for un received blocks . Once a block is got, the peer cancels the block request from its neighbor peers in order to decrease the bandwidth wastage due to repeated downloads.

#### 2.4.2 Neighbor Selection Strategy

The peer selection strategy employs four main mechanisms namely tit-for-tat (TFT), optimistic unchoking (OU), anti-snubbing, and upload only. The above mechanisms not only aims at improving the downloading capabilities of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

contributing peers but also provides a way to penalize the free riders.

**2.4.2.1 TFT Policy**

The tit-for-tat policy aims at uploading the peers preferably those who offers the best downloading rates. A leecher gives preference to the best three neighbors who provides the best downloading rate and chokes others. The leecher reevaluates the downloading rate from all peers every 10 seconds who are sending a data to it. If some other peer offers a better downloading rate, the leecher will replace the peer with the smallest downloading rate and unchoke the better peer. This mechanism aims mainly at encouraging contributors and punishing the free-riders .

**2.4.2.2 Optimistic Unchoking Policy**

This policy provides an opportunity for finding out the peers that can provide higher uploading rate. Optimistic unchoking is done once in every 30 seconds.

**2.4.2.3 Anti-snubbing Policy**

Whenever a peer notices that the time has elapsed due to the poor download rates, the leecher assumes that the peer is'snubbed' and it does not allow any further upload through the regular unchoke.

**2.4.2.4 Upload Only Policy**

According to this policy the seeds prefer to upload the peers that offers better uploading rates once a peer completes downloading.

**2.4.3 Incentive Mechanisms**

The scheme of incentives [4] are introduced in a peer-to-peer system in order to convince the strategic peers to cooperate with each other. There are variety of incentive schemes and some of there are discussed here.

**2.4.3.1 Monetary Payment Schemes**

In monetary payment schemes, users are required to pay in some form of virtual currency in order to get the specific services from other peers but it suffers form scalability problem.

**2.4.3.2 Reciprocity-Based Schemes**

In reciprocity-based schemes, peers maintain a behavior histories of other peers for decision making processes.

**2.4.3.3 Credit based incentives**

In a credit–based method a peers earn a credits whenever a users download the files from them, and spend credits when they download files from others.

**2.4.3.4 Reputation based incentives**

In a reputation based systems, peers earn a reputation based on past behavior. The reputation increases when they forward the packets and reputation decreases when they don't forward packets. The main drawback of a reputation scheme is that the peers with a low reputation are avoided in path selection and may be punished.

**2.4.3.5 Contribution based incentives**

In a contribution based method the peers will reciprocate based on the received volume of contribution from others.

**2.4.3.6 Effort based incentives**

In the effort based policy, peers reciprocate based on the effort put up by the other peers relative to its upload capacity. It treats both the slow and fast peers evenly. It is considereded to be a both fairer and efficient system dfue to the increase in overall download performance.

**2.4.3.7 Team based incentives**

The Team based method organize the peers of similar upload bandwidth in teams. Team members satisfy the needs of data download within their team and only perform optimistic unchokes when needed. It provides an incentives for contributing peers to join a team by rewarding them with improved download rates and also discourages free-riders by means of limiting the number of optimistic unchokes.

**2.4.4 Based on Service Capacity**

**2.4.4.1 Chunk Based Switching**

In the chunk-based switching scheme, the file is divided into many small chunks and an user downloads the chunks sequentially one at a time. Whenever a user downloads a chunk from its current source peer, the user randomly selects a new source peer and connects to it to retrieve a new chunk. By this way of switching the source peers based on chunk can reduce the correlation in service capacity between chunks and hence reduce the average download time.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

#### 2.4.4.2 Time Based Switching

In the time based switching , a peer changes one of its servicing peer with the lowest upload capacity every 10 seconds with a hope of finding some peers offering higher service capacity.

#### 2.4.4.3 Choke Based Switching

Choking based switching includes a preemptive choke at the client level that enables a client to identify a choke threshold independently that can be used for an intelligent escape from poor performing servers. Choke algorithm not only decreases the download duration in single client and multi client scenarios but also increases the prediction accuracy of the file download duration.

## 3 RELATED WORKS

B. Cohen, inventor of a Bittorrent has illustrated a systematic introduction to the BitTorrent system in [4]. The paper includes the protocol description of the BitTorrent, the Bittorrent architecture and the built in incentive mechanisms. In addition, many work reporting the efficiency and the popularity of BitTorrent [8, 9].

Neglia, et al., [10] conducted a BitTorrent based measurement study to get a clear view of a large scale internet file distribution based on multitrackers, and distributed hash tables (DHT). Both multi trackers and DHTs help in balancing the load on the server. The study was made over 1,400 trackers and 24,000 DHT nodes for about two months. The study findings are : Firstly, A measurement study based on a single tracker provides an improved availability compared to the measurement based on multitrackers. Secondly, The correlated failures reduce the chance of availability. Thirdly , Multi trackers reduce the overlay connectivity formed by the peers. Lastly, the use of DHT ensures high response latency in peer queries.

D. Qiu and R. Srikant, [9] is the pioneer in constructing the mathematical model for the BitTorrent system. The paper proposed a fluid model to describe the population evolution of seeds and leechers in the BitTorrent system. There are many papers in BitTorrent systems regarding an analytical-based [8] and measurement-based studies [7].

Xu, et al., [11] proposed the deployment of helpers in order to improve the BitTorrent network performance. Helpers are high-bandwidth, high-connection and controllable super nodes employed for modeling, simulating and analyzing. The study summarizes that the helper deployment improves the system performance when there is no constraint in maintaining overall uploading bandwidth.

Izal et al., [8] presented a measurement based BitTorrent studies with the help of different torrent tracker logs. The study revealed a a constant high average download rate. LRF is considered to be an effective policy as the nodes are ready to upload s peers as soon as it obtains a few blocks. TFT manages a positively correlated upload/download times.

Li, et al., [12] proposed a biased, optimistic, unchoking mechanism called PicBou, which unchokes the neighbors optimistically when it has a rarely replicated file pieces among internal neighbors. It helps the peers to get a missing blocks and share the blocks among the neighbors once the blocks are downloaded into an ISPs, thereby reducing the content distribution time and bandwidth consumption. The performance of the proposed mechanism was evaluated through the simulations which results in 20 percent reduction in content distribution time and 20-35 percent reduction in inter-ISP traffic.

Zhang, et al., [13] studied the sampling bias for BitTorrent and introduced the taxonomy of sampling bias sources. The study illustrated that the measured BitTorrent components have the significant results in terms of bias measurement. Based on the results, the rules where formulated to reduced sampling bias.

Blond, et al [14] performed an extensive simulations with more than 10000 BitTorrent clients for evaluating the impact of high locality. The inter-ISP traffic and peers download completion time were evaluated. The results have shown that the overhead can be reduced significantly by a small number of inter-ISP connections. Two methods namely Round Robin and Partition Merging, were introduced for utilizing a real torrents in the internet.

Y. M. Chiu and D. Y. Eun [18] proposed a distributed algorithm in order to reduce the average file download time by means of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

reducing the negative impacts of two factors namely temporal correlations and heterogeneity in service capacity.

Lehrfeld et al. [19] proposed a switching algorithm which includes a preemptive choke at the client level that enables a client to identify a choke threshold independently that can be used for an intelligent escape from a poor performing servers.

## 4 SIMULATION SETTINGS

The nodes joining the network is modeled to study the performance of bandwidth distribution of a peer to peer network based on a probabilistic approach. Most of the connections are assumed to be an ADSL with a low upload bandwidth value than the download bandwidth value. Table 1 bives the uplink/downlink bandwidth value to be used in the simulation. A file size of 500 Mb with the block size of 256 Kb is used in the simulation with an initial seed having a bandwidth of 1024 Kbps. There can be 200 active nodes at a given time. The file size of 500 Mb is splitted into a 256 Kb blocks resulting in a total of 2000 blocks. The maximum number of neighbour for each node is limited upto 5. The efficiency of a bittorrent is evaluated by the mean utilization of uplinks or downlinks over time. Utilization point can be calculated by an aggregate traffic flow ratio of all uplinks/downlinks to the aggregate capacity of all system uplinks. The maximum data can be served when the network uplinks are saturated. The uplink utilization is considered to be a key performance determinant though the downlink utilization is an important access link asymmetry .

*Table 1 : Available Bandwidth*

| Downlink Kbps | Uplink Kbps | Nodes percentage |
|---|---|---|
| 384 | 128 | 30 |
| 896 | 128 | 20 |
| 64 | 64 | 30 |
| 1920 | 128 | 20 |

### 4.1 Metrics

The efficacy of the BitTorrent is evaluated by means of the metrics namely link utilization and fairness.

**Link utilization:** The utilization at any point in time is calculated as the mean ratio of the aggregate traffic flow on all uplinks/downlinks to the aggregate capacity of all uplinks/downlinks in the system that is the ratio of the actual flow to the maximum possible.

**Fairness:** When an uplink bandwidth is scarce , it is important to measure the fairness ratio.It is measured by the ratio of peer's upload/download utilization at any point in the torrent.

### 4.2 Simulator Used :

The discrete-event simulator namely OCTOSIM models the peer activities like peer joins, peer leaves, block exchanges and bittorrent mechanisms in detail. The network model considers a downlink and an uplink bandwidth for each node to model the asymmetric networks. The bandwidth settings are set up appropriately in order to delay the blocks exchanged by nodes. The computation of block transmission is expensive so that we limit the maximum no of nodes to 1000. The number of nodes can be extended as time goes on.

### 4.3 Simulation Parameters :

The following table shows the simulation parameters to be assumed for the simulation

*Table 2 : Simulation Values*

| Default Simulation Parameters | |
|---|---|
| File Size | 500MB |
| Block Size | 256 KB |
| No of Blocks | 2000 |
| Concurrent Upload | 5 |
| Initial Seed | 1 |
| Arrival Pattern of Peer | Flash crowd |
| Tracker Updation period | 30 s |

## 5 RESULTS

The performance of the proposed architecture is tested by increasing the nodes linearly by a factor of 100 while running an experiment. disassociate when file download is complete.

Two block selection policies including random policy and Local Rarest First (LRF) are considered to choose blocks from neighbors. Table 3 and Figure 4 shows mean upload utilization over time.

Journal of Theoretical and Applied Information Technology
10th November 2013. Vol. 57 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 3: Upload and Download utilization*

| No of nodes | Mean Upload utilization | | Mean Download utilization | |
|---|---|---|---|---|
| | LRF | Random | LRF | Random |
| 100 | 95.2 | 97.2 | 37 | 37.3 |
| 200 | 97.3 | 98.24 | 42 | 40.83 |
| 300 | 97.4 | 98.26 | 43.5 | 42.5 |
| 400 | 97.8 | 98.26 | 43.93 | 43.78 |
| 500 | 97.9 | 98.28 | 44.02 | 43.91 |
| 600 | 98 | 98.3 | 44.21 | 43.98 |
| 700 | 98.1 | 98.32 | 44.2 | 44.14 |
| 800 | 98.1 | 98.37 | 44.76 | 44.36 |
| 900 | 98.4 | 98.35 | 44.81 | 44.8 |
| 1000 | 98.4 | 98.37 | 44.82 | 44.8 |

*Table 4 : Fairness Ratio*

| Nodes Count | Fairness Ratio (LRF VS Random | |
|---|---|---|
| 100 | 2.57 | 2.61 |
| 200 | 2.32 | 2.41 |
| 300 | 2.24 | 2.31 |
| 400 | 2.23 | 2.24 |
| 500 | 2.22 | 2.24 |
| 600 | 2.22 | 2.24 |
| 700 | 2.22 | 2.23 |
| 800 | 2.19 | 2.22 |
| 900 | 2.20 | 2.21 |
| 1000 | 2.20 | 2.21 |

Figure 2 reveals that random policy provides better upload utilization than the LRF policy. Utilization between LRF and Random policy converges as nodes increase. Regarding download utilization LRF performance and random policy are nearly the same. Figure 3 reveals that the download utilization varies between 40% and 45% showing a very good download utilization in both policies. Fig 4 shows that the random policy provides better fairness than LRF policy for increasing no of nodes.

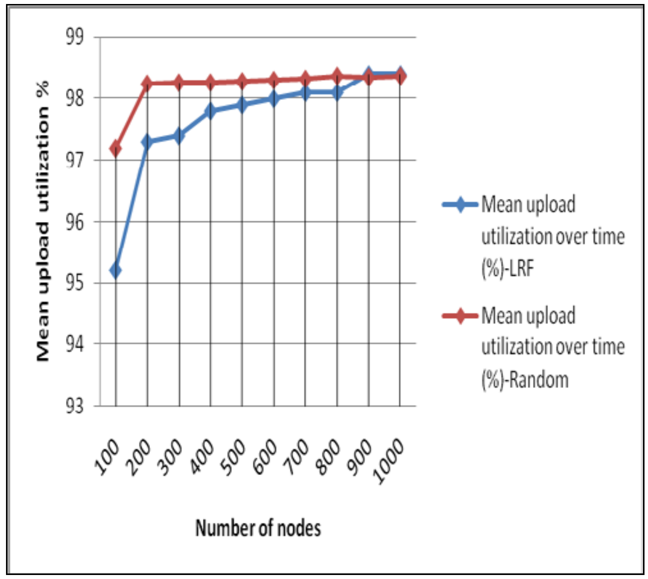



*Fig. 2: Mean Upload Utilizatio*

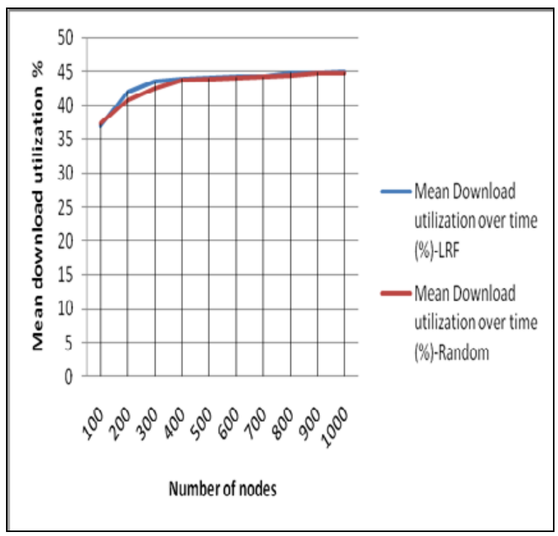



*Fig 3 : Mean Download Utilization*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

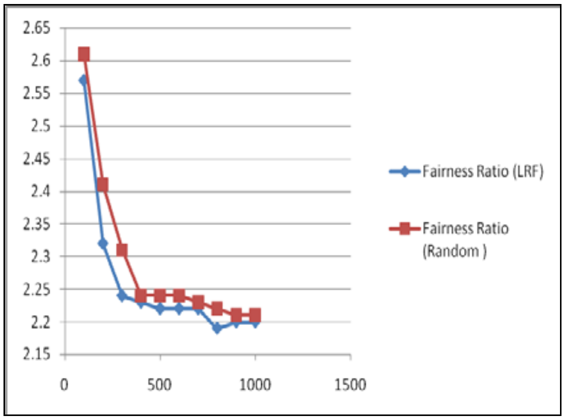



*Fig 4 : Fairness Comparison*

## 6 CONCLUSIONS

This paper simulate a bit torrent like network and study the performance under a scenarios including LRF policies and random policy for picking up blocks for download from neighbors. The major consideration is the percentage of active nodes , link utilization and fairness. Simulations were undertaken with nodes varying between 100 and 1000. The results reveals that random policy outperforms than LRF policy in terms of upload utilization and download utilization performed by two policies are almost same. Future work focus on rarely downloaded files with few leechers and fewer seeds.

## *REFERENCES*

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

[2] Rodrigues, R., & Druschel, P. (2010). Peer-to-peer systems, *Communications of the ACM*, 53(10), 72-82.

[2] Schulz, Hendrik; Klaus Mochalski (2009), *Internet Study 2008/2009* , Leipzig, Germany.

[4] Cohen, B. (2003, June), Incentives build robustness in BitTorrent, In *Workshop on Economics of Peer-to-Peer systems* (Vol. 6, pp. 68-72).

[5] Xu, J., Wang, X., Zhao, J., & Lim, A. O. (2012), I-Swifter: "Improving chunked network coding for peer-to-peer content distribution, *Peer-to-peer networking and applications*, 5(1), 30-39.

[6] Piatek, M., Krishnamurthy, A., Venkataramani, A., Yang, R., Zhang, D., & Jaffe, A. (2010, April), "Contracts: Practical contribution incentives for P2P live streaming", *In Proceedings of the 7th USENIX conference on Networked systems design and implementation (pp. 6-6). USENIX Association.*

[7] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips, A Measurement Study of the BitTorrent Peer-to-Peer File-Sharing System,*Technical Report PDS-2004-003*, Delft University of Technology, The Netherlands, April 2004.

[8] Izal, M., Urvoy-Keller, G., Biersack, E. W., Felber, P. A., Al Hamra, A., & Garces-Erice, L. (2004). Dissecting bittorrent: Five months in a torrent's lifetime, *In Passive and Active Network Measurement* (pp. 1-11). Springer Berlin Heidelberg.

[9] Qiu, D., & Srikant, R. (2004), Modeling and performance analysis of BitTorrent-like peer-to-peer networks , *ACM SIGCOMM Computer Communication Review*, 34(4), 367-378.

[10] Neglia, G.; Reina, G.; Honggang Zhang; Towsley, D.; Venkataramani, A.; Danaher, J.; , Availability in BitTorrent Systems, INFOCOM 2007. *26th IEEE International Conference on Computer Communications. IEEE* , vol., no., pp.2216-2224, 6-12 May 2007.

[11] Ke Xu; Yahui Yang; Tao Chen; , Improving BitTorrent Network's Performance via Deploying Helpers, *Embedded and Ubiquitous Computing, 2008*, EUC '08. IEEE/IFIP International Conference on , vol.2, no., pp.507-512, 17-20 Dec. 2008

[12] Zhenyu Li; Gaogang Xie; , Enhancing Content Distribution Performance of Locality-Aware BitTorrent Systems, *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference* , vol., no., pp.1-5, 6-10

[13] Zhang, B., Iosup, A., Pouwelse, J., Epema, D., & Sips, H. (2010). Sampling bias in BitTorrent measurements. *In Euro-Par 2010-Parallel Processing* (pp. 484-496). Springer Berlin Heidelberg.

[14] Le Blond, S., Legout, A., & Dabbous, W. (2011). Pushing bittorrent locality to the limit. *Computer Networks*, 55(3), 541-557.

[15] Adar, E., & Huberman, B. A. (2000). Free riding on gnutella, *First Monday*, 5(10-2).

[16] Legout, A., Urvoy-Keller, G., & Michiardi, P. (2006, October). Rarest first and choke algorithms are enough, *In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (pp. 203-216). ACM.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[17] Yuh-Ming Chiu and Do Young Eun , Minimizing file download time in peer to peer networks , IEEE/ACM Transactions on Networking, VOL. 16, NO. 2, APRIL 2008

[18] Michael R. Lehrfeld and Greg Simco , Choke-based Switching Algorithm in Stochastic P2P Networks to Reduce File Download Duration, IEEE 2010