A Fuzzy Approach For Piece Selection In Bit Torrent Like Peer To Peer Network

* M. Padmavathi, ** R. M. Suresh

* Assistant Professor, Department of Information Technology, RMD Engineering College, PO box 601 206, Kavaraipettai, Chennai; E-mail: padmavathim2000@gmail.com
** Principal, Jerusalem College of Engineering, PO box 600 100, Velachery Main Road, Pallikaranai, Chennai

Abstract: There is no clear distinction between client and server nodes in a peer-to-peer (P2P) network. Network nodes (or peers) requiring resources also contribute resources to the network. Such an arrangement ensures that more resources like CPU, storage, and bandwidth are available for other peers when more nodes enter the network. BitTorrent is a popular and scalable peer-to-peer file distribution mechanism successful in distributing large files expeditiously and efficiently without creating problems for the origin server. Measurement studies revealed that BitTorrent achieved excellent upload utilization, but also led several questions regarding utilization in settings other than those measured, fairness, and choice of BitTorrent's mechanisms. This paper proposes a novel technique for block selection based on fuzzy logic. The proposed technique uses the best features of Random piece selection and Bandwidth based piece selection ensure fairness. The proposed technique is evaluated based on the link utilization through various scenarios.

[M. Padmavathi, R. M. Suresh. **A Fuzzy Approach For Piece Selection In Bit Torrent Like Peer To Peer Network.** *Life Sci J* 2013;10(4s):482-489] (ISSN: 1097-8135). http://www.lifesciencesite.com. 74

Keywords: Peer-to-Peer (P2P) network, BitTorrent, Download time, Performance

1. Introduction

Peer-to-Peer (P2P) networks are a type of computer network which are characterised by self-organizing, with symmetric communication and distributed control [1]. The P2P networks are contrast of the traditional client-server networks. The server of traditional client-server networks requires large link capacity to feed its clients, whereas in P2P networks resources of each peer are pooled for the common good. P2P applications are used on the internet as its inherent characteristics make it useful in content distribution as compared to client-server architecture. In P2P networks, as peers are both client and server simultaneously there is no clear distinction between them. Hence, it takes less time to distribute data when compared to a client-server method. Peer-to-peer (P2P) systems provide powerful infrastructure for large-scale distributed applications, like file sharing resulting in their popularity. P2P accounts for 43% of Internet traffic. Among P2P systems, BitTorrent is highly prevalent as it accounts for more than 50% of all P2P traffic [2], delivering large data volume from origin servers to clients. However, BitTorrent differs from classic P2P application as the peers are responsible for sharing content. [3].

When connecting with neighbours, each node downloads and uploads blocks from and to neighbours. When a peer completes a file download it becomes a seed leading to a flood like file spreading among peers. When additional peers join the swarm, it increases the chances of a successful download. Reduced hardware and bandwidth resource cost, redundancy against system problems and reduced dependence on the original source are some of its advantages. BitTorrent protocol employs local rarest first (LRF) technique to choose blocks for downloading from neighbours. Least replicated blocks among neighbours are picked by LRF. The block diagram of a P2P network is seen in Figure 1.

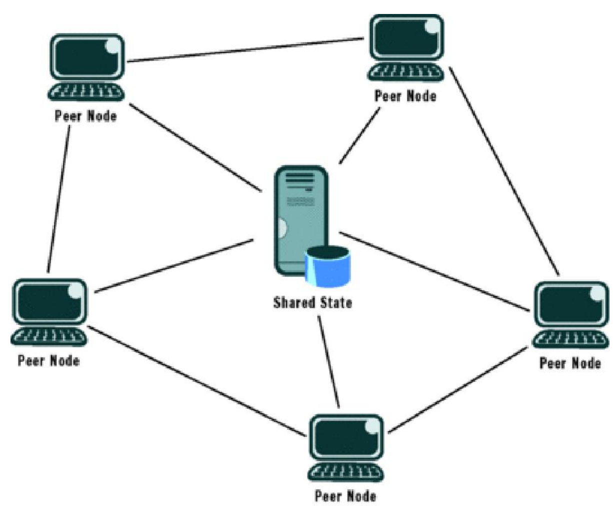

Figure 1: A P2P network

BitTorrent [4] is a P2P file sharing protocol ensuring fast and efficient distribution of large files through upload bandwidth control of downloading peers. BitTorrent divides the files into equal sized blocks allowing nodes to join a 'swarm' of peers to upload/download simultaneously. The file to be downloaded is named 'torrent', divided into blocks ranging in size from 32 to 256 KB. A swarm is a peer group downloading/uploading, BitTorrent can upload torrent blocks before the torrent is totally downloaded which ensures that it is better than other P2P protocols. Camouflaging of request-response latency is achieved through dividing blocks into sub-blocks to enable pipelining of requests [5]. A central 'tracker' records every network node periodically updating information about nodes joining/leaving a torrent.

Three features make BitTorrent successful. First, it breaks a complete file into blocks disseminating it by sending blocks instead of the complete file. Second, BitTorrent uses LRF block selection algorithm to disseminate blocks, ensuring that users will download the rare block among neighbors. It was proved that LRF algorithm can enlarge service capacity and prevent last block problems [6]. Third, is the rate-based Tit-for-Tat (TFT) unchoking scheme which discourages freeriders in the BitTorrent system successfully, as freeriders will be repeatedly choked when they fail to provide uploads to other users.

Simulation-based approaches allow understanding and deconstructing BitTorrent performance. Live internet measurements like tracker logs [7], or live Torrent participation are not enough to study performance metrics. As configuration parameters of BitTorrent mechanisms can't be controlled or alternate methods cannot be incorporated, simulators are used for studying BitTorrent performance. The simulator helps to study and model peer activity including joining/leaving a torrent and block exchanges. All mechanism like LRF, TFT can be incorporated in detail. Asymmetric network access is modelled by associating a downlink/uplink bandwidth for every node. Simulations can be carried out on live data or else it is possible to hypothetically model scenarios leading to better knowledge of peer to peer networks.

2. Literature Survey

Liao, et al., [8] proposed a mathematical model which is able to predict the average file download delay in a heterogeneous BitTorrent-like system, accurately. The proposed model is common and is derived with least assumptions and needs least system information. A flexible token-based approach for BitTorrent-like systems is proposed and applied to trade off among the entire system performance and fairness to excessive bandwidth users, by correctly setting its parameters. To predict the average file download delays in the token-based system, and illustrate the application of the model to define the parameters of the scheme which accomplishes a target performance/fairness for which the proposed mathematical model is extended.

Deaconescu et al., [9] introduced a framework for BitTorrent performance evaluation which is implemented to evaluate and compare the recent real world BitTorrent implementations. The main issue considered is evaluating and comparing different executions between several BitTorrent clients, each utilizing varied algorithms and performance optimization methods. To output the transfer status data and many logging information the clients have been well instrumented as the framework is completely automated.

Bharambe, et al., [10] introduced a study of BitTorrent that is simulation-based which destroyed the system's design and estimates the effect of its core mechanisms both independently and by grouping, under a diversity of workloads on the entire system performance. Many significant metrics comprising file download time, peer link utilization, and fairness among peers in terms of volume of content served are focused in the estimation. BitTorrent performance is validated from the results obtained that it is nearly-optimal in terms of download time, and uplink bandwidth utilization excluding the intense conditions. The illustrations reveal that during the presence of high bandwidth peers, the low bandwidth peers are able to download much more than they upload to the network. It is observed that to prevent unfairness, the rate-based TFT policy is not efficient. To enhance the fairness efficiently basic modification to the tracker and a stricter, block-based tit-for-tat policy is introduced.

Pouwelse, et al., [11] focused on four main issue namely availability, integrity, flash-crowd handling, and download performance for measurement study of BitTorrent. The goal of this work is to provide better knowledge about a real P2P system which possesses accurate mechanisms to facilitate millions of users, for modeling P2P systems essential measurement data should be provided, and in these systems the designing issues are recognized efficiently.

Chiang, et al., [12] proposed an Interest-Intended Piece Selection (IIPS) algorithm in order to reduce the last piece problem when sustaining a steady cooperation among peers. When downloaded, every IIPS peer favours pieces that have the interest intended in it will maximize the probability of being interesting to its cooperating peers. Under harsh conditions, less occurrences of piece loss is accomplished by IIPS is illustrated from its simulation results and in terms of higher piece diversity it slightly outperforms the BitTorrent's rarest-first algorithm.

Esposito, et al., [13] proposed BUTorrent for enhancing the downloading time. Because of the need of global knowledge and the overlay dynamics the initial phase is delicate in a content distribution (file sharing) scenario. This phase is able to cause delays in attaining a steady state, hence maximizes file download times which is unwise piece dissemination. A new class of scheduling algorithms at the seed based on a proportional fair scheme is devised and on a real file sharing client it is implemented. In addition to simulation results the proposed file sharing client (BUTorrent) is validated as the average downloading time of a standard file sharing protocol enhanced by 25%. The theoretical upper bounds over the achievements that the scheduling strategies may accomplish are also provided.

SeungChul Han et al., [14] examined the problem of selecting a server for parallel download in overlay content-distribution networks. In a hypercube-like overlay network, a node-selection method produces the optimal server set based on the worst-case link stress (WLS) criterion is proposed. The scaling to a huge system is facilitated by the algorithm due to its efficacy as it does not needs topology collection or routing information or network measurement. In many fields especially against the random selection approach it has performance benefits. The level of congestion at the bottleneck link is initially reduced which is similar to improving the high-throughput. Secondly, in terms of the total number of links utilized and the total bandwidth usage of the network resources is consumed in less quantity. Mostly a low average round-trip time to selected servers is guided. Thirdly, it facilitates more data exchange with neighbor nodes that are a content-distribution system's main objective.

Eger, et al [15] proposed two pricing based bandwidth trading schemes for fast file distribution. Explicit price information and the download rates from other peers as the price were the basis of the proposed algorithms. The proposed algorithms were compared with BitTorrent in both static and dynamic scenarios. The proposed algorithms achieved better fairness when compared to BitTorrent. Nam, et al [16] investigated the effect of introducing BitTorrent nodes in different networks. The OPNET simulation result demonstrated that the end-to-end delay increases due to the P2P traffic on networks. The simulations also illustrated that the congestion is reduced due to the tracker and choke algorithms even though network performance degrades. Xia, et al., [17] surveyed the Bittorrent performance, and summarized the findings of the study. Improvements to BitTorrent's mechanism available in the literature were also studied and summarized. Based on these studies, the authors suggested improvements to further improve the performance. Table 1 summarizes the improved mechanisms for BitTorrent.

Table 1: Improved Mechanisms for BitTorrent

Author	**Proposed Method**	**Result**	**Issues Unsolved**
Liao, et al., [8]	Predict the average file download delay	Accomplishes a target performance/fairness	Bandwidth fairness is not accomplished
Bharambe, et al., [10]	Block-based tit-for-tat policy	Enhance fairness	Bandwidth fairness is not accomplished
Chiang, et al., [12]	Interest-Intended Piece Selection (IIPS) algorithm	Less occurrences of piece loss	Fairness to peer is not achieved
Esposito, et al., [13]	BUTorrent	Enhance downloading time	Peer fairness not achieved
SeungChul Han et al., [14]	Worst-case link stress (WLS)	Scaling to a huge system, Reduced congestion, less bandwidth usage	Peer fairness not achieved
Eger, et al [15]	Two pricing based bandwidth trading schemes	Fast file distribution, better fairness	Frequent piece loss

This paper aims to investigate the impact of Bit-Torrent's mechanisms and parameters on overall performance. First, it should be known whether BitTorrent can keep all system uplinks active and used fully. Full use means, optimal mean download time and also optimal load on the origin server. There are many reasons why uplink use might be less. One is that nodes take independent downloading decisions with regard to file blocks, hence neighbours might get a similar set of blocks decreasing their utility. How the blocks get replicated is also governed by peers used block-selection policy. The policy used by BitTorrent, LRF, may not also be optimal. It should also be found out if other policies work equally well and under what workloads policy choice becomes crucial. TFT constrains peer connections resulting in a scenario where a node decides not to serve its peer, in spite of having useful blocks to serve.

Also key BitTorrent mechanisms are dependent on parameters like peers number which interacts with each node and maximum permitted uploads. These mechanisms interact differently and performance could be influenced by specific parameter settings. In this paper, the BitTorrent is simulated for evaluating the effect of core mechanisms on the system under different workloads. This paper proposes a fuzzy approach where fairness mechanism is integrated with block selection policy to ensure that fairness is based on the way the node behaves in the network.

3. Methodology

In a bid to study P2P network, it is planned to model the nodes joining the network through a probability based approach for uplink/downlink bandwidth distribution. It is assumed that most connections to be ADSL with upload bandwidth lower than download bandwidth during modelling. A 100Mb file with 256 Kb block size is used for simulation, with one seed having a bandwidth of 1024 Kbps. A total of 1000 active nodes are available at any one time. The number of neighbour for each node is limited to 5. BitTorrent efficiency is evaluated through mean utilization of peers' uplinks/downlinks over time. Aggregate traffic flow ratio on all uplinks/downlinks to the aggregate capacity of all system uplinks/downlinks is the utilization at any point. This provides a basic idea of actual flow to the maximum. The system serves data to the maximum when all network uplinks are saturated. Though downlink utilization is important, access links asymmetry makes uplink as the performance's key determinant.Theoretically, it is assumed that the nodes have unlimited capacity but in actuality nodes require large local buffer to store the data.

The proposed architecture's performance is tested by linearly increasing the nodes by a factor of 200 in each experiment, under the assumption that all nodes joining a network will disassociate themselves when downloading of the total file is completed. The proposed technique is compared with two policy scenarios - random policy and Local Rarest First (LRF) for choosing of blocks from neighbours. The rarest first algorithm works as follows. Each peer maintains a list of the number of copies of each piece in its peer set. It uses this information to define a rarest pieces set. Let m be the number of copies of the rarest piece, then the index of each piece with m copies in the peer set is added to the rarest pieces set. The rarest pieces set of a peer is updated each time a copy of a piece is added to or removed from its peer set. Each peer selects the next piece to download at random in its rarest pieces set.

Some policies are used to modify the performance of the rarest first algorithm. First, the random first policy, a peer chooses randomly for the next requested piece when it has downloaded less than 4 pieces. On downloading a minimum of 4 pieces, it shifts to the rarest first algorithm. Thus random first policy permits a peer to download its first pieces faster than with the rarest first policy, this is required as the peer needs some pieces to respond for the choke algorithm.

In the bandwidth based technique, the users are classified as high bandwidth users and low bandwidth users. The average download delay for each group and the whole system is evaluated. A token system is introduced to study the performance of the high and low bandwidth users. The tokens are used to trade blocks by the users. Token tables are maintained by users, which contain the information of the neighbor's tokens. When the user uploads U_{up} bytes to a neighbor, the neighbor's token reduces by $T_{down}U_{up}$. And neighbor's token value increases by $T_{up}U_{down}$ with a download of U_{down} bytes by the user from their neighbor. This ensures fairness to all the nodes actively participating in the network.

ALGORITHM
INPUT: Available Bandwidth, Available block.
OUTPUT: Piece selection technique
STEP 1: Find the degree to which the input belongs in the membership function.
STEP 2:Fuzzify the input
STEP 3:Aggregate the outputs
STEP 4:Defuzzify

DIAGRAM

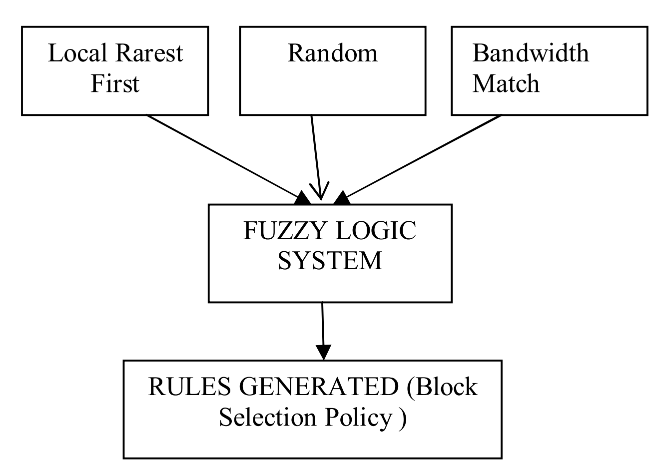

Figure 2: Flowchart of the Proposed Method

The proposed technique uses a combination of LRF and Bandwidth based technique based on the availability of blocks and the cooperation of nodes during downloads. The input linguistic variable selected for this purpose is the available bandwidth of the node and the availability of a specific block across the entire network. The input variables are fed into the fuzzy logic system, and the rules are generated. Based on the rules the proposed method chooses either LRF or bandwidth method during downloads. The following Figure 2 shows the flowchart of the proposed method.

4. Result and Discussion

During modelling, we assume most of the connections to be ADSL with upload bandwidth lower than the download bandwidth. The simulation parameters are tabulated in Table 2, and the uplink and downlink bandwidth used is shown in Table 3.

Table 2: Simulation Parameters

Parameter	**Value**
Connection	ADSL
File Size	100 Mb
Block Size	256 Kb
Bandwidth	1024 Kbps
Number of neighbour for each node	5
Total number of Nodes	200, 400, 600, 800, 1000

Table 3: Bandwidth Availability with Each Node

Downlink Kbps	**Uplink Kbps**	**% of nodes with the said bandwidth**
384	128	30
896	128	20
64	64	30
1920	128	20

The input linguistic variable selected for this purpose is the available bandwidth of the node and the availability of a specific block across the entire network. The Membership Function (MF) uses a bell curve in the midrange as shown in Figure 3 for both the input linguistic variable. The following equations are used to perform the logical operations, where union of sets represents 'OR' and conjunction represents 'AND' of the logical operations:

Assume 2 fuzzy sets A and B which are defined in space X, the union of the two sets forms set C which is C⊂X, with membership function defined as:

$$\mu_c(x) = Max\left[\mu_A(x), \mu_B(x)\right] \quad (1)$$

where $\mu_A(x)$ is membership function of set A and $\mu_B(x)$ is membership function of set B.

Similarly, for conjunction of sets A and B,

$$\mu_c(x) = Min\left[\mu_A(x), \mu_B(x)\right] \quad (2)$$

For Implication 'IF…THEN..' the membership is defined as:

$$\mu_{A\to B}(x, y) = \mu_R(x, y) = min\left[\mu_A(x), \mu_B(x)\right] \quad (3)$$

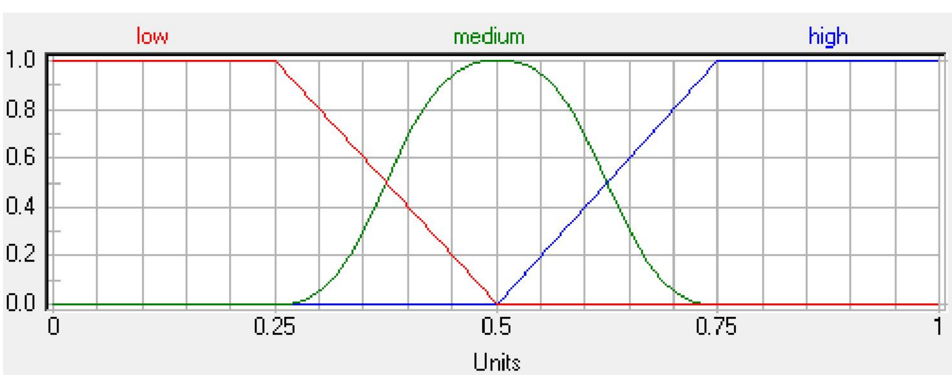

Figure 3: Input membership function for Block available and Bandwidth

The definition points of the input membership function are shown in Table 4.

Table 4: Definition Point of The Input Membership Function

Term Name	Shape/Par.	Definition Points (x, y)		
low	linear	(0, 1) (1, 0)	(0.25, 1)	(0.5, 0)
medium	S-Shape/0.50	(0, 0) (0.75, 0)	(0.25, 0) (1, 0)	(0.5, 1)
high	linear	(0, 0) (1, 1)	(0.5, 0)	(0.75, 1)

The output membership function for the policy to be chosen is shown in Figure 4 and table 5 shows the output definition point.

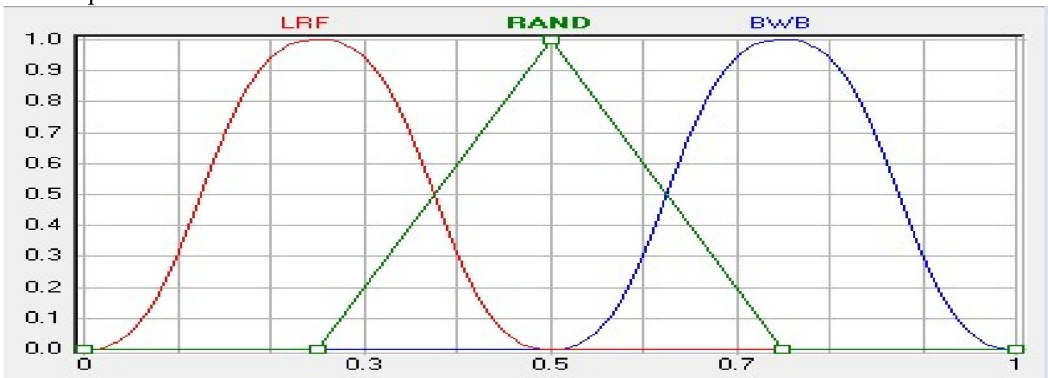

Figure 4: The policy selected based in input MF

Table 5: Definition Point of Output Membership Function

Term Name	Shape/Par.	Definition Points (x, y)		
LRF	S-Shape/0.50	(0, 0) (1, 0)	(0.25, 1)	(0.5, 0)
Rand	linear	(0, 0) (0.75, 0)	(0.25, 0) (1, 0)	(0.5, 1)
BWB	S-Shape/0.50	(0, 0) (1, 0)	(0.5, 0)	(0.75, 1)

The control strategy of the fuzzy logic system is contained in the rule blocks. Each rule block restricts all rules for similar context. The 'if' part of the rules, define the condition for which the rules are designed and the 'then' part gives the output of the fuzzy system for that condition. The importance of the rule is weighed using degree of support (DoS). The processing of the rules starts with calculating the 'if' part. The operator type of the rule block determines which method is used. The rules generated are shown in Table 6.

Table 6: Rules Generated

IF		THEN	
Bandwidth Available	**Block Available**	**DOS**	**Policy**
low	low	0.25	LRF
low	low	0.5	Rand
low	low	0.75	BWB
low	medium	0.25	LRF
medium	low	0.5	LRF
medium	low	0.5	Rand
medium	low	0.75	BWB
medium	medium	0.5	LRF
high	low	0.25	LRF
high	low	0.5	Rand
high	medium	0.75	LRF

Simulations were carried out after modifying the OCTOSIM simulator. Simulations were carried out with 200,400,600, 800 and 1000 nodes in the network. 10 runs were simulated. Table 7 shows the mean upload and download utilization of the network for the best run.

TABLE 7: Utilization within the network for best run

Number of nodes	Mean upload utilization over time (%)			Mean Download utilization over time (%)		
	LRF	Random	Fuzzy	LRF	Random	Fuzzy
200	97.3	98.24	**99.01**	42	40.83	**41.75**
400	97.8	98.26	**99.05**	43.93	43.78	**44.01**
600	98	98.3	**99.12**	44.21	43.98	**44.82**
800	98.1	98.37	**99.12**	44.76	44.36	**45.08**
1000	98.4	98.37	**99.1**	44.82	44.8	**45.34**

From table 7, it can be seen that the performance of the proposed system using the best of the three techniques provides improved utilization of the bandwidth for both uploads and downloads. This is consistent with the number of nodes joining the network increases linearly. Figure 5 and figure 6 shows the mean upload and download utilization in the network.

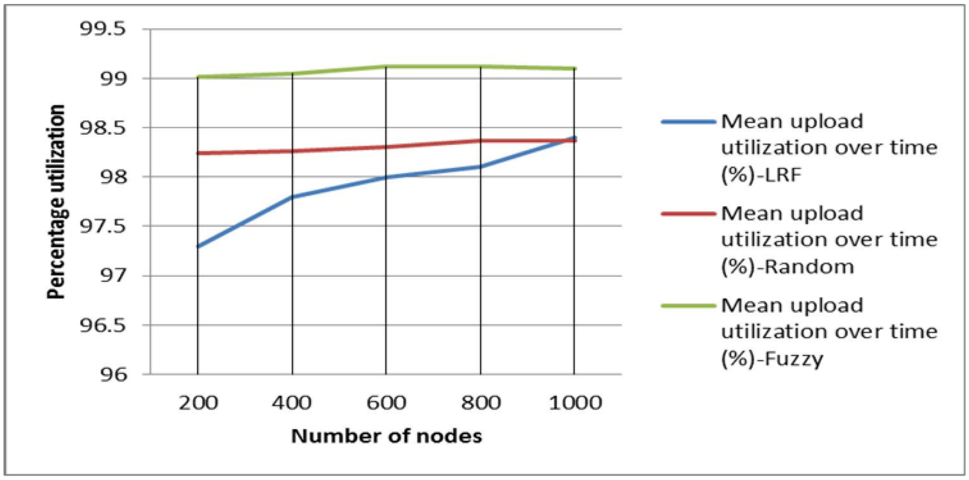

Figure 5: Mean upload utilization for the best run

The improvement in the download utilization time is not very significant in the proposed system; however the mean download utilization is higher than other systems tested.

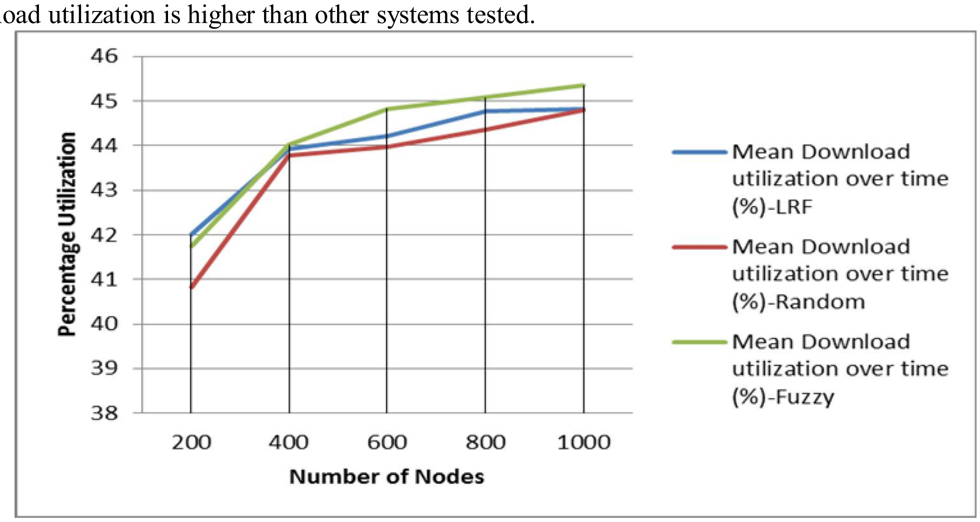

Figure 6: Mean download utilization for the best run

The number of nodes joining the network is increased from 200 to 1000. The graph represents the performance of the network with the increase in the number of nodes. They join network during 10s periods and stay till the download is completed. Figure 5 shows the mean upload utilization which is the average across all the nodes. The upload utilization achieved is very high, above 99% for varying number (200 to 1000) of nodes in the network. These high values of uplink are the indication that the network performance is optimal with respect to mean download time.

4. Conclusion

In this paper, it is proposed to simulate a bit torrent like network and study the same under various scenarios. The scenarios considered include the LRF policies, random policy and the proposed fuzzy based policy for picking up blocks for download from its neighbor. Another important consideration was the number of nodes used in the network. Simulations **are** carried out with network containing 200 nodes to 1000 nodes for various bandwidth considerations. The result obtained shows the upload utilization by the proposed technique is better than random policy and LRF policy. In this work, it was assumed that bandwidth remains constant during the entire process which may not be necessarily true in real time system. Since the network parameters are additive in nature, the proposed solution is NP-Complete. Further work needs to be done to address the NP complete problem.

REFERENCES

1. Risson, J., & Moors, T. (2006). Survey of research towards robust peer-to-peer networks: search methods. Computer Networks, 50(17), 3485-3521.
2. Parker, The true picture of peer-to-peer filesharing. URL: http://www.cachelogic.com/, Accessed February 2007.
3. Schulz, Hendrik; Klaus Mochalski (2009): Internet Study 2008/2009". Leipzig, Germany
4. "BitTorrent," http://bittorrent.com.
5. Cohen, B. (2003, June). Incentives build robustness in BitTorrent. In Workshop on Economics of Peer-to-Peer systems (Vol. 6, pp. 68-72).
6. Bharambe, A. R., Herley, C., &Padmanabhan, V. N. (2005). Analyzing and improving BitTorrent performance. Microsoft Research, Microsoft Corporation One Microsoft Way Redmond, WA, 98052, 2005-2003.
7. Pouwelse, J. A., Garbacki, P., Epema, D., & Sips, H. J. (2004). A measurement study of the BitTorrent peer-to-peer file-sharing system. Technical Report PDS-2004-003, Delft University of Technology, The Netherlands.
8. Liao, W. C., Papadopoulos, F., & Psounis, K. (2007). Performance analysis of BitTorrent-like systems with heterogeneous users. Performance Evaluation, 64(9), 876-891.
9. Deaconescu, R., Rughinis, R., & Tapus, N. (2009, April). A bittorrent performance evaluation framework. In Networking and Services, 2009. ICNS'09. Fifth International Conference on (pp. 354-358). IEEE.
10. Bharambe, A. R., Herley, C., & Padmanabhan, V. N. (2005). Analyzing and improving BitTorrent performance. Microsoft Research, Microsoft Corporation One Microsoft Way Redmond, WA, 98052, 2005-2003.
11. Pouwelse, J., Garbacki, P., Epema, D., & Sips, H. (2005). The bittorrent p2p file-sharing system: Measurements and analysis. Peer-to-Peer Systems IV, 205-216.
12. Chiang, J. L., Tseng, Y. Y., & Chen, W. T. (2011). Interest-Intended Piece Selection in BitTorrent-like peer-to-peer file sharing systems. Journal of parallel and distributed computing, 71(6), 879-888.
13. Esposito, F., Matta, I., Bera, D., & Michiardi, P. (2011). On the impact of seed scheduling in peer-to-peer networks. Computer Networks, 55(15), 3303-3317.
14. Han, S. C., & Xia, Y. (2009). Optimal node-selection algorithm for parallel download in overlay content-distribution networks. Computer Networks, 53(9), 1480-1496.
15. Eger, K., & Killat, U. (2008). Bandwidth trading in bittorrent-like p2p networks for content distribution. Computer Communications, 31(2), 201-211.
16. Nam, K. K., & Xu, Y. J. (2011). Analysis of BitTorrent Protocol and Its Effect on the Network.
17. Xia, R. L., & Muppala, J. K. (2010). A survey of BitTorrent performance. Communications Surveys & Tutorials, IEEE, 12(2), 140-158.
18. Joung, Y. J., Wong, W. T., Huang, H. M., & Chou, Y. F. (2012). Building a network-aware and load-balanced peer-to-peer system for range queries. Computer Networks.

2/26/2013