Efficient Sharing of Application using Fairness Data in Distributed Environment

K.Kalpana[#1], Deepak Lakshmi Narashima[*2] & S.Kavitha[#3]
[# 1]Computer Science & Engineering, School of Computing,
SASTRA University, Tirumalaisamudram, Thanjavur-613401, Tamilnadu, India.
[1]kalpanakcse@gmail.com
[# 3]Computer Science & Engineering, School of Computing,
SASTRA University, Tirumalaisamudram, Thanjavur-613401, Tamilnadu, India.
[3]kavilfrnd@gmail.com
[* 2]Computer Science & Engineering, School of Computing,
SASTRA University, Tirumalaisamudram, Thanjavur-613401, Tamilnadu, India.
[2]deepak@it.sastra.edu

***Abstract* -Torrent is used to download the same data by applying the bit torrent communication protocol. Some of the common terminologies used in torrent are peers, leechers, seeders and trackers. The peers are sometimes referred as leechers, the leecher is somebody who is currently downloading the file, the seeder is somebody who is currently uploading a file and tracker is the one who keeps track of all the activities of leecher and seeder. This paper mainly focuses on file sharing in cheap and effective peer to peer system, which suffers due to slow uploading and downloading and bandwidth lacking fairness. In this paper we report on experiments conducted to pin point the cause to increase the Bandwidth in uploading the data. And so we are proposing a new system using the k-nearest neighbour algorithm to ensure fairness. KNN algorithm is the simplest of all machines learning algorithms. By using this algorithm the objects are classified and grouped by neighbour's voting providing rendering the shortest path among varying neighbours. K-is the smallest integer. If K=+1, then the object is grouped to the class of nearest neighbours. Thus our proposed methodology results in bandwidth fairness by increasing the performance of downloading and providing bandwidth in a uniform manner.**

Keywords: Torrent, Fairness, Peer-to-PeerSystem, Bandwidth, KNNAlgorithm

I.INTRODUCTION

Bit torrent is a peer to peer file sharing technology, meaning that when you download a file over bit torrent you are not getting it from a central server; you are getting it from other users like you who have downloading file, or even just part of it. The features of the bit torrent are uTorrent in Windows platform, Deluge in all platform, Transmission in Mac OSX/*nix, rTorrent in *nix and Vuze in all platforms. In that uTorrent is light-weight bit torrent client, the Deluge is also a light weight and cross-platform in bit torrent client and then the Transmission system is empty and clear origin bit torrent node on large download blocks and an increment great in set, rTorrent is a text based bit torrent node it runs on Linux and Unix systems and Vuze is also free and cross platform application in java it is also known as Azures.

Fairtorrent is also known as bit torrent. In that free riders get less amusing downloading times for leading peers. If the peer to peer system, the fair bandwidth assignation where the fairness we mean that the peer receives bandwidth equal to what it contributes, the system would be able to guarantees a certain level of performance for contributing peers. In that some fairtorrent they used deficit based distributed algorithm to ensure fairness. In that word deficit means, deficit = Number of the bytes sent to the neighbours - Number of the bytes received by the neighbours. In that fairness many approaches are to be used, they are Tit for Tat (TFT), Block based and Prop Share. TFT is the most popular file sharing system and Block based is used by Bit Tyrant and Prop Share is used to prove better performance over TFT by Proportional response algorithm. In this client's waste much time in discovering better peer to exchange data width.

The main disadvantage is the overall uploading and downloading performing lower data bandwidth. In block based uploading and downloading data cannot be processed by assuming bandwidth rate. In this paper, the bandwidth is not distributed in the uniform manner and so downloading performance is decreased. The allocation of file sharing in the router bandwidth cannot be known in advance and the strategic way of the file sharing can reduce the bandwidth level in the uploading the data.

To overcome these problems: our aim is to increase the upload and download bandwidth in fairtorrent making use of fair bandwidth. We present the K-Nearest neighbour algorithm, whenever we have to classify a new point, by finding K-nearest neighbours from the training data. The distance is to be calculated by the fairness data.

The rest of the paper deals with following sections: in that related works some of the torrents are to be discussed, system design describing the overall architecture of our proposing system, system execution explaining the working concept of proposed algorithms, evaluation part gives information about performance analysis, conclusion proves how we succeeded in achieving our scope and future work.

II. RELATED WORK

Many researches are being done in torrent file sharing system some of them are as follows: one of the papers discussed about the power, volume and geometry constraints has been written by Obulapathi N.Challa et... al[1]., The cube sat torrent techniques are used to increase the downlink and uplink speeds of large file. In that large files are distributed into different pieces and placed in the cluster containing cube sat. From that different cluster, different pieces of files are downloading simultaneously.

The work discussed by Damini Kapse et... al.[2]. is to authentic .torrent file they introduced client Torrentzshare, a search and recommendation system. They are some of client are recommended as Torrent swapper, Vuze and Tribler.

The paper by Abdellatif olama et...al[7], discusses on flash crowd situation is used to improve average download time and utilization of the cellular bandwidth.

Haiyang Wang and Jiangchuan Liu[3] developed a smart detection mechanism. This mechanism used to decrease the erosion of peers and cut through ISP traffic to be minimized.

Alex Sherman et...al.[6], modified chocking algorithm and proposed bandwidth allocation algorithm. Using this algorithm, they increase the information measure for downloading and cross through the file shift time by using leecher torrent. It improved the fairness in peer to peer system using fairtorrent deficit based algorithm. It avoids the pitfalls and inaccurate bandwidth.

Fairtorrent was compared with bit torrent Azureus, PropShare, Bit Tyrant and proved that, it provide high degree of fairness (Mihai capote and Nazareno Andrade[5]). In this paper, they used mechanism for resource allocation. File sharing is used to maximize throughput. To find max-min fair allocation, video-stream was used.

Zengbin Zhang et.. al.[12] predominantly guaranteed the efficiency and fairness of file sharing. It is provided by the strong correlation among the seed and peer.

Another paper discussing Fluid model based analysis used to download multiple torrents and several multitorrent downloading scenario and multi file torrent downloading scenario are illustrated. In that multi torrent downloading analysis two things are multi torrent concurrent downloading and multi torrent sequential downloading based on fluid model.Next the multi file torrent downloading comes under collaborative multi file torrent sequential file transferring founded on fluid model.

III. SYSTEM DESIGN

The proposed system is in fairtorrent we use fair bandwidth for all clients to increase the upload and download bandwidth. The overall system architecture showing uploading and downloading file Fig: 1 from torrent is as follows:

The design issues to be discussed are authentication to fairtorrent, upload and download, fair bandwidth, KNN and Fairtorrent implement in bit torrent.

In Fig: 2 authenticating a fairtorrent are only the authorized user can login into the system and the unauthorized user cannot enter into the system.

If any new user wants to login means they must be registered. Then only the user is certified as authorized user or else unauthorized user.

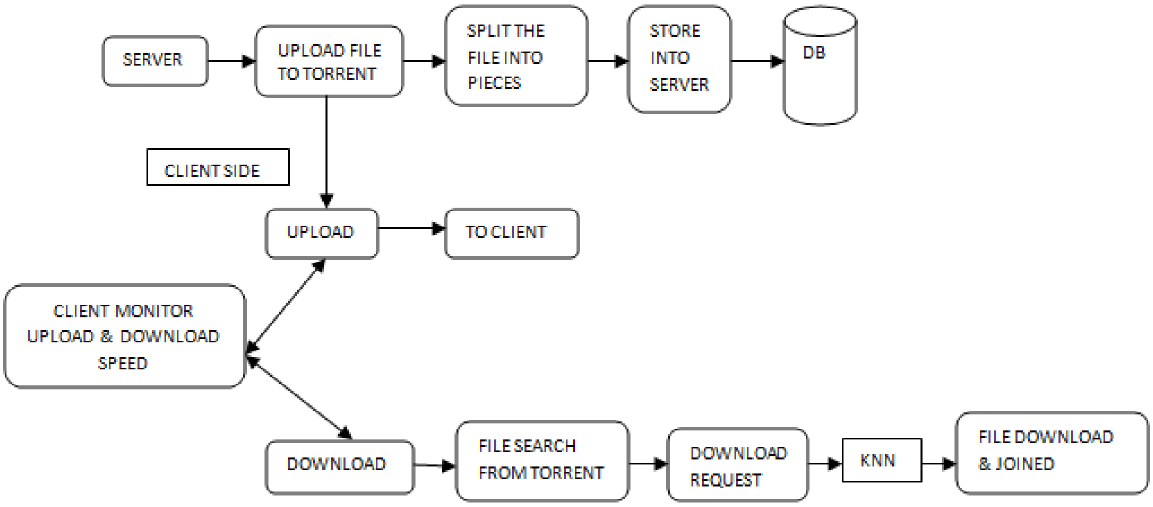


Fig 1: Overall architecture of proposed system

The following diagram shows the authentication system of a torrent.

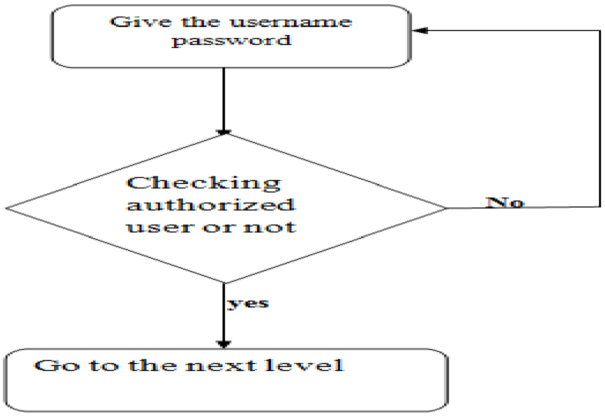


Fig 2: Authentication System

Next design issue fig: 3 is upload and download part, in which the user can download files which are already uploaded, after completion of a particular download based on the download rate the new active set of peers has been uploaded. Bit torrent has a feature of updating the list of peers after every download. Every addition of peer is for chocking and deletion of a peer is for unchoking.

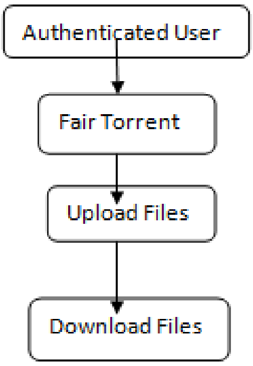


Fig 3: Uploading and downloading

The fair torrent implementing in bit torrent system has been explained in fig: 4. the concept of fairtorrent which proves an efficient performance over bit torrent is, in fairtorrent a fair data exchange has been implemented using a fully distributed algorithm providing service beyond bit torrent. The mechanism taking place in fairtorrent is whenever a client uploads a data block it automatically converges to the single interchange place of its peers, without quantifying or proclaiming these rates.

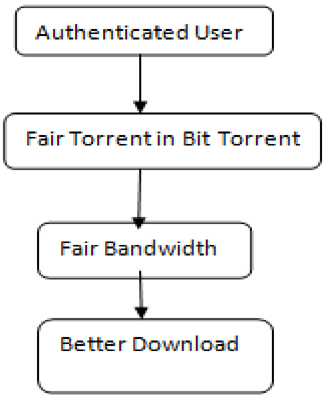


Fig4: Fairtorrent implement in Bit torrent

Then next module is to download the next packets for the current Torrent file ID, the Fairtorrent searches for the nearest node again by kth nearest node algorithm and ensure fairness in that node, and begins download by providing fair bandwidth. Finally after downloading all packets from the server which are in a distributed fashion are united and download as a file to the requested active user.

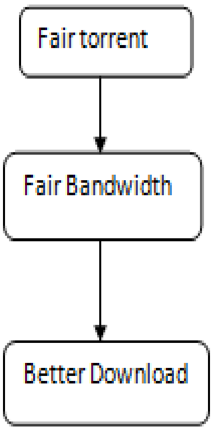


Fig 5: Fair Bandwidth

Using our proposed algorithm K-Nearest Neighbour algorithm Fig 5, the user can download files only by uploading files (i.e.) the active site of peers that a client uploads to modify each round based on the measurement of their download rates. The files is uploaded and divided into number of packets. During file upload to the server, the file information regarding path, size of the file and torrent id are created. The fairtorrent, search for the nearest neighbour node algorithm for that torrent id and download the packets to the active clients.

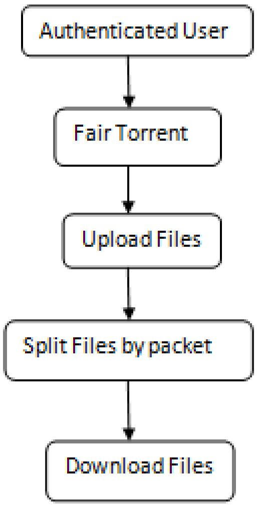


Fig 6: KNN

IV. SYSTEM EXECUTION

The main discussion of the working part is to increase the bandwidth and we can use the K-Nearest neighbour algorithm and round trip time algorithm to calculate the speed of time. First we have to find the nearest neighbour to download the files we need from our nearest neighbour.

By using the torrent id of the files present in server we can search for it and download the packets. In KNN algorithm we make use of varying parameters such as node, networks, file, file packets and bandwidth of the execution part of the system. In that algorithm we find the nearest neighbour node of the process. We search the nearest neighbour and the whole packet has been splitted into many small packets and at last we join all the files whiles downloading in client.

Algorithm for K-Nearest Neighbour:

```
Networks (n) = upload (file) {
if (upload(s) == n)
upload (file) = file packets;
n= nmax;
  if (n! = null in nmax)
     {
          n=n+1;
      }
bmax = round trip time;
    if (n=bmax)
      {
          Find n;
       }
download (file) = file packets in n;
if ((file packets) =file)
{
join (file packets);
}
```

Then making use of the round trip time algorithm we are calculating the round trip time which is referred as length of the time and the constant value is to be S, then the two nodes i and j and transmission to be discussed are below is,

Algorithm for Round Trip Time

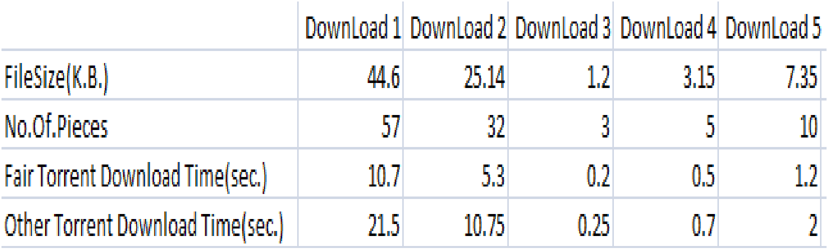
```
if (n (t)! = null)
{
        if (n(i) to n(j))
         {
Round trip time =(S*old RTT) + (1-S)*new RTT)...... (0<=S<1);
calculate RTT for ack;
        }
}
```

Then we have to calculate the speed using min-max algorithm by varying parameters such as speed, files and maximum bandwidth is given by,

```
if (upload (file packets)! =file)
{
n=n+1;
node= bmax;
upload(s) =n;
}
```

V. EVALUATION

In the evaluation part we are discussing the performance of the older torrent with the performance of the fair torrent which is our proposed system. Comparing with other torrent mechanisms which are available already, the fairtorrent download time is very low which is illustrated by the following graph.



	DownLoad 1	DownLoad 2	DownLoad 3	DownLoad 4	DownLoad 5
FileSize(K.B.)	44.6	25.14	1.2	3.15	7.35
No.Of.Pieces	57	32	3	5	10
Fair Torrent Download Time(sec.)	10.7	5.3	0.2	0.5	1.2
Other Torrent Download Time(sec.)	21.5	10.75	0.25	0.7	2

Table 1: Older Torrent Performance

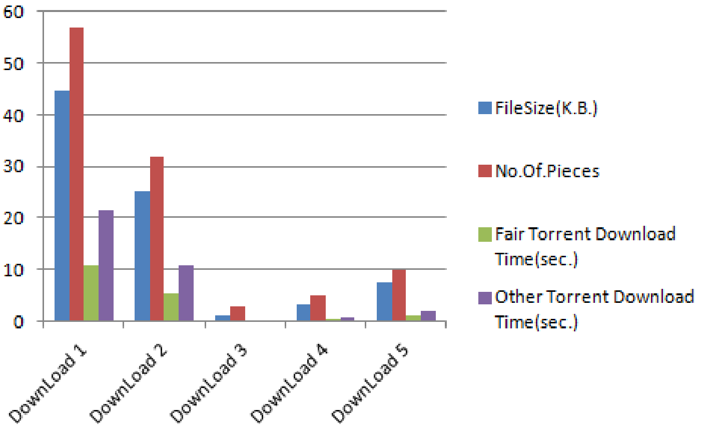
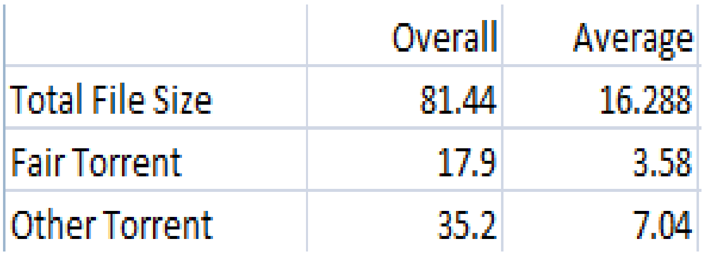


Fig 7: Older Torrent Analysis

As a concluding part of my proposed system performance, am showing how the proposed architecture proves us with a better performance when compared to other torrent mechanisms. The experiment has been conducted for varying file size packets and the time taken for their transmission has been noted. By analysing these recorded results the performance has been proved as efficient. The newer performance analysis graph is as follows:



	Overall	Average
Total File Size	81.44	16.288
Fair Torrent	17.9	3.58
Other Torrent	35.2	7.04

Table 2: Proposed Torrent Performance

Thus the fairtorrent performance has been increased by 49.14772727%, is the overall improved performance range of the system.

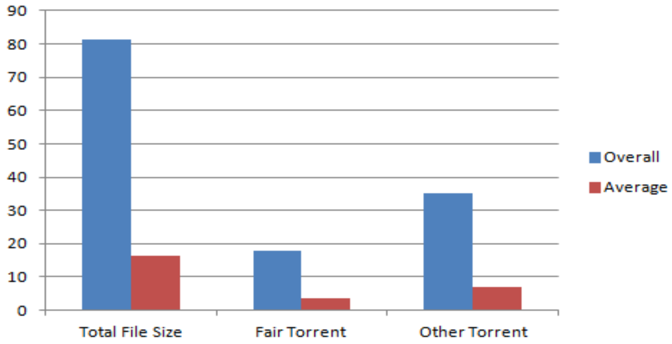


Fig 8: Proposed Torrent Analysis

VI. CONCLUSION

Fairtorrent has been developed to avoid the pitfalls of the previous approach deficit based distributed system and so we are making use of my proposed algorithm k-nearest neighbour. The algorithm is to search the nearest neighbour, and increasing the performance level of the system.

In Fair torrent the user is able to download the files in a distributed environment from different client in better time .It is possible because here the user is able to identify the torrent id of each file in a list and requesting each part of the file. To download each part of file from different user the fair torrent is able to find the kth nearest node with fair bandwidth from the user .As a result the user is able to download his file at 49.14%better download time.

VII. FUTURE WORK

Increasing the bandwidth provisioning for the client will never have a intensity level. We can show our proposed system is showing a better performance than other systems and we can show to some extent our proposed methodology shows a increased performance in bandwidth utilization. And the download time has been considerably minimized comparing to older methodologies. As a future enhancing approach for this model we can make use of streaming while transfer of data files.

VIII. REFERENCES

[1] Obulapathi N.Challa and Dr.Janise McNair, “Cube Sat Torrent: Torrent like Distributed Communications for Cube Sat Satellite Clusters”, Proc. IEEE transaction on parallel and distributed system, 2013.

[2] Damini Kapse, Ritesh Mittal, Manali Jiwtode and Dr. Prachi Gharpure, “TORRENTZSHARE Search and Recommendation of torrents”, Proc. 2012 International Conference on communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India.

[3] Haiyang Wang and Jiangchuan Liu, “Exploring Peer-to-Peer Locality in Multiple Torrent Environment”, Proc. IEEE transaction on parallel and distributed system, VOL. 23, NO. 7, July 2012.

[4] Jaeyoung Choi, Jinyoung Han, Taejoong Chung, Eunsang Cho, Taekyoung Kwon and Yanghee Choi,” Bandwidth Allocation for Bit Torrent under Multi-Torrent Environments”, IEEE transaction on parallel and distributed system, 2011.

[5] Mihai Capota, Nazareno Andrade, Tamas Vinko, Flavio Santos, Johan Pouwelse and Dick Epema, “Inter-swarm resource allocation in Bit Torrent communities”, Proc. IEEE transaction on parallel and distributed system, 2011.

[6] Alex Sherman, Jason Nieh and Clifford Stein,” FairTorrent: Bringing Fairness to Peer-to-Peer Systems”, CoNEXT’09, December 1–4, 2009, Rome, Italy.

[7] Abdellatif Olama, Amira Thabet and Amnal El-Nahas, “Cell-Torrent: A Hybrid File-Sharing Model For Wireless Networks”.

[8] M. Sirivianos, X. Yang, and S. Jarecski, “Dandelion: Cooperative content distribution with robust incentives,” in Proc. USENIX, Jun. 2007.

[9] F.Wu and L. Zhang, “Proportional response dynamics leads to market equilibrium,” in Proc. 39th Annu. ACM STOC, Jun. 2007, pp. 354–363.

[10] Ye Tian, Di Wu and Kam-Wing Ng, “Analyzing Multiple File Downloading in Bit Torrent”, Proc. the 2006 International Conference on Parallel Processing (ICPP'06).

[11] Abdellatif Olama, Amira Thabet and Amnal El-Nahas, “Cell-Torrent: A Hybrid File-Sharing model for Wireless Networks”, Proc. IEEE transaction on parallel and distributed system.

[12] Zengbin Zhang, Yao Li, Yang Chen, Pei Cao, Beixing Deng and Xing Liy, ”Understand the Unfairness of Bit Torrent”, Proc. IEEE transaction on parallel and distributed system.

[13] [Online]. Available: http://www.bittorrent.com.