IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. III (Jan. 2014), PP 74-79
www.iosrjournals.org

Reducing Cross-ISP Traffic in P2P Systems Using Adaptive Search Radius

Rohit Ranjan[1], Arup Bhattacharjee[2]
[1]*National Institute of technology silchar, India*
[2]*National Institute of Technology silchar, India*

Abstract: *Peer to Peer communication has become very popular these days .This popularity and increase in P2P traffic has given birth to many internet traffic management problems for service providers. One of these problems is high download traffic_ of P2P file sharing application for long network distances ,as current applications do not pay attention to network topology while selecting peers for sharing data .To reduce this trans network traffic various solutions have been suggested but all of them lead to compromises for upload rate, or file availability ,or external infrastructure on the internet to support the solution. This papers suggest a modified version of Adaptive search radius algorithm for BitTorrent to solve the problem , at minimum compromise to efficiency of file sharing applications .This papers shows how BitTorrent protocol react to modified adaptive search radius , as it is the most favored application for peer to peer file sharing .*
Key words: *P2P, Peer to Peer, Adaptive Search Algorithm, BitTorrent.*

I. INTRODUCTION

Peer to peer communication is the most popular way of sharing content on internet between users. The best thing is , it do not require any infrastructure ,so do not add any cost to the publisher of the data. Peer to peer file sharing application account for 50% to 80% of the Internet download traffic[1].
Totally different from client sever file sharing applications it is much better user oriented method for sharing data . There is no need of server ,by the publisher of the data. Most popular data is easily available on p2p network and at much higher download rate.

But due to its rising popularity and its user centric approach of data sharing has resulted into a big traffic management problem for the ISP's. As p2p file sharing application do not use topological information of the network they produces very high data traffic which needs to travel long distance on the internet, which leads to high cross ISP traffic.

These type of communication where peers are at large distance from each other in terms of hop count , consumes ISP bandwidth of originating ISP as well as on those ISP who fall between them. This is inefficient way of communication as it produce high load on ISP network causing congestions , which are neither good for ISP's nor for the users.

ISP's pay to there parent ISP's according to there total data traffic flow which is both download and upload traffic. So those ISP who fall between to peers and don't own them also have to pay for the data transfer on there network , which adds operational cost to them. Due to high traffic of p2p applications , these cost adds up to very high amount.

This traffic also use there bandwidth which affect other users using other application on the network. ISP's are under constant pressure due to competition ,to provide good speed to there users and so they are bound to reduce these high download traffic which neither originate nor destined to there users. These ISP's can't totally bock p2p application as there user's also use this applications. To be fare they use other techniques to shape, throttle or even block the traffic so as to reduce this trans network traffic to certain acceptable level. But these traffic shaping devices affect to efficiency of p2p systems .So to solve the problem certain internal solution within the p2p file sharing protocol is required.

In the past few years various author have suggested different solutions to tackle this problem or to have a fare arrangement in the protocol which could be a win-win position for both users and internet service providers.

Biased neighbor selection [2] mechanism focuses on locality ,if data is available in the neighborhood we should select local peers over the far ones for downloading, this will localize the traffic and reduce global traffic .In BNS, locality is judged in the terms of ISP Hops , required to reach destination peers.BNS is a good way to tackle this problem but require huge ISP's intervention for finding out information about packets and network topologies.

TopBT [3] mechanism is more independent in terms on infrastructure requirement and can significantly improve the overall internet traffic management. It suggest dual use of link hops as well as (autonomous Systems)AS hops to find out closeness information of the peers . It also focus on monitoring connections rates to decide with which node it should communicate with.

Nearby neighbor Selection [4] method is a modified version of BNS . It suggest that instead of choosing N-k nodes from neighbor ISP's ,K =1 should be used, as for K>=1 the download time does not changes much.

UTAPS [5] is topology aware solution to this trans network traffic problem .It also suggest to select peers within small hops counts using knowledge of topology. UTAPS is a tracker supported solution in which for each new peer joining the network ,tracker needs to use Traceroute method to find out topological knowledge about the peer.

Some authors also suggested other locality policies like tracker locality where tracker has the responsibility to keep on updating topological information and use that to provide closest peerset to the peer ,asking for peerset [6].

Another way to reduce cross ISP traffic is through gathering low cost updates from content distribution networks about the topology , and exploit that information to help peers decide peer selection in a much better and efficient way [7].

Network aware Peer selection algorithm [8] can calculate distance between the peers using traceroute results and than classify peers into three categories – local, intra ISP and inter ISP.

Adaptive Search Radius [9] so far is best solution as it do not require any external help from tracker or ISP's for its peer selection algorithm . Like Nearby Neighbor Selection it also use least hop count method but suggest dynamic search radius which keep on changing depending upon current piece availability . ASR is able to reduce the trans network traffic but do not provide optimal solution as it is hindered by choking algorithms . Seeders will try to unchoke those peers who have high upload rate no matter how far they are on the network. This paper try to use ASR and change choking algorithm of bitTorrent to a certain level so that optimal radius could be used without giving rise to free riding problem.

This paper is organized as follows. Section 2 gives a detail study of BitTorrent Protocol working ,its file structure and algorithms used in decision making. Section 3 is study of Adaptive Search radius Algorithm [9]. Proposed changes to optimistic unchoking algorithm is there in section 4. Section 5 deals with simulation part and have graphs to show the outcomes. Section 6 describes the future work to be carried out.

II. The BitTorrent Protocol [10]

BitTorrent is a protocol for distributing files. It identifies content by URL and is designed to integrate seamlessly with
the web. Its advantage over plain HTTP is that when multiple downloads of the same file happen concurrently, the downloaders upload to each other, making it possible for the file source to support very large numbers of downloaders with only a modest increase in its load.

To share a file using BitTorrent you have to publish somewhere in the web a .torrent file containing information
such as the file name, its length, hashing info and the tracker address. This special node of the overlay network allows the peers to know other peers that are already interested to the file. Upon request of a peer, the tracker replies providing a set of 50 random nodes which may have a part or the whole file. This set of neighbors can increase if the local peer is in the peerset (i.e. the set of neighbors returned by the tracker which are interested to download the file) of another node.

Initially the node who decides to share something has
the whole file, then to correctly share a file there must be at least a copy of every piece belonging to it in the network, otherwise the file's download will be never completed. A node with all the pieces of a file is called seeder otherwise is called leecher. The set of peer sharing the same file is called swarm.

2.1 File structure

Inside a peer, a file is split in pieces of 256 KB each and every piece is split in 16 sub-pieces (or blocks) (every block is 16 KB). If the download of a piece crashes, the piece has to be entirely re-downloaded. Splitting the file in pieces enhances the precision in the estimate of the download time and allows a fast download of the block also from peer with low bandwidth. Different blocks belonging to the same piece can be downloaded from different peers. After the download of a piece has been completed, the peer must send a HAVE message to its neighbors.

2.2 Chocking algorithms

This algorithm manages the downloading rates in the network. Every node tries to maximize its downloading rate
and does this by using the "tit-for-tat" philosophy. This technique, born from the games theory, states that a player (in our case a peer) will be cooperative with another one only in the case this one will do the same. It is a principle of collaboration that benefits components interested to the exchange of informations.

The chocking is a temporary refuse of uploading pieces to a peer. The main idea is to provide a correspondence between upload and download traffic, achieving connections in both the directions.

2.3 Unchoking algorithm

There are several criteria a good choking algorithm should meet. It should cap the number of simultaneous uploads for good TCP performance. It should avoid choking and unchoking quickly, known as 'fibrillation'. It should reciprocate to peers who let it download. Finally, it should try out unused connections once in a while to find out if they might be better than the currently used ones, known as optimistic unchoking.

The currently deployed choking algorithm avoids fibrillation by only changing who's choked once every ten seconds. It does reciprocation and number of uploads capping by unchoking the four peers which it has the best download rates from and are interested. Peers which have a better upload rate but aren't interested get unchoked and if they become interested the worst uploader gets choked. If a downloader has a complete file, it uses its upload rate rather than its download rate to decide who to unchoke.

For optimistic unchoking, at any one time there is a single peer which is unchoked regardless of its upload rate (if interested, it counts as one of the four allowed downloaders.) Which peer is optimistically unchoked rotates every 30 seconds. To give them a decent chance of getting a complete piece to upload, new connections are three times as likely to start as the current optimistic unchoke as anywhere else in the rotation.

3. Adaptive Search Radius

Adaptive search radius [11] is a peer selection algorithm which client peers can use to decide from which peer it needs to communicate and asks for piece. First step is to discover list of peers that share the file in which client is interested. In normal Torrent application , client will try to download from as many serving peers as it can irrespective of which is close or far from him .ASR basic strategy is to restrict downloading to local peers only. To do that ,ASR uses search radius measured in terms of router hops .Search radius is the current number of hops, client peer message needs to travel to communicate the current farthest peer. To find out and decide which peer is closest and which one is far . We can use UDP /TCP ping message to find out the router hops required to reach the serving peer. This distance metric will be calculated for all peers. ASR also uses another metric called file availability which is numbers of peers, within the current radius ,sharing the rarest file piece. File availability metric is used to dynamically adjust the search radius. The protocol has max_Availability and min_Availablity as two fixed parameters.

ASR requires to initially set radius to a value that is suitable enough to cover entire Internet (e.g. 64 hops). Contact all the peers inside this radius to get information about the file status of all the peers. If the file availability metric is higher than min_Availabilty than , reduce radius by one hop and re check the condition. Keep on reducing the radius until file availability comes equal to min_Availabilty or slightly above it.

If file availability becomes less than min_Availabilty , than increase search radius by one , keep on repeating until file availability come equal or slightly above min availability.

```
CalculateSearchRadius(file, currentSR)
currentAvailability ← FileAvailability(file, currentSR)
if currentAvailability > MAXAVAILABILITY
    then while FileAvailability(file, currentSR − 1) ≥ MINAVAILABILITY
            do currentSR ← currentSR − 1
elseif currentAvailability < MINAVAILABILITY and AllPeersContacted(file, currentSR)
    then currentSR ← currentSR + 1
return currentSR
```

Fig. 1 ASR Radius Calculation [9]

III. Proposed Modification for Optimistic Choking

The current adaptive search radius is fine for emule as it does not have any specific peers selection algorithm for choosing which peer to serve from all the request ,which a peer gets.

But BitTorrent implements choking and optimistic choking algorithm to deal with free riding problem. As in the choking algorithm peers with high upload rate are unchoked first and with low upload rate have to wait .So ASR radius in such condition depends upon ,from how many seeders a client peer is unchoked and how far are they. ASR search radius will not be optimal one, there is chance to further reduce the radius and localize the traffic if we could make serving peer to unchoke close peer first over far ones. But this property will under mine the solution for free riding problem.

So, we need to have careful tradeoff between free riding problem and heavy trans network traffic problem . If we try to reduce one , other will appear.

Serving node in BitTorrent has to unchoke 4 peers from all the request it get. 3 are unchoked on the basis of upload rate ,and last one randomly from rest of the peers list. We modified the unchoking algorithm and change the optimistic unchoking on the basis on locality , i.e 4th peer to be unchoked will be the closest peer in the request list of the serving peer. If N is the number of peers to be unchoked by serving peer (4 for BiTorrent), we used N-K nodes to be unchoked on the basis of upload rate and K nodes in the basis of closeness , where 0<K<=N. After analysis of the outputs for different K values , we suggest K=1 should be used, as it provide better spreading rate of the file in the network as well as provide moderate upload rate to the seeders without hurting tit for tat policy

IV. Simulation And Evaluation

This paper answers above proposal by extensive simulation study. We have tried to simulate all relevant mechanisms in BitTorrent clients. Our simulator provides delay to the packets with respect to the distance between the peers. The simulator models concurrent uploads, calculation of download rates from neighbors, the choking / unchoking algorithm's. This simulation mainly focuses on analytical model of real world overlay.

To evaluate the effect of locality during unchoking, We tried to simulate the real network traffic scenario on PeerSim simulator[12] , which is a event driven simulator perfect for p2p network simulations. We simulated a network of 700 nodes with a constraint of 450 nodes to form a swarm in our network at a time.

Table 1. Characteristics Of Simulation.

File_size	100 MB
Max_swarm_size	450
Peerset_size	50
Max_growth	20
Newer_distr	90%
Seeder_distr	10%
remove	10
add	10
step	10000 ms

To get the simulation more realistic , system keeps on adding 10 new peers in the swarm and at the same time remove 10 random peers from it after every 10 sec. In all addition and removal operations of peers, 90% of peer will be leechers and 10% will be seeders. File which is transferred between the peers is of size 100MB ,divided into 390 pieces each piece(256KB) further divided into 16 blocks of 16KB.

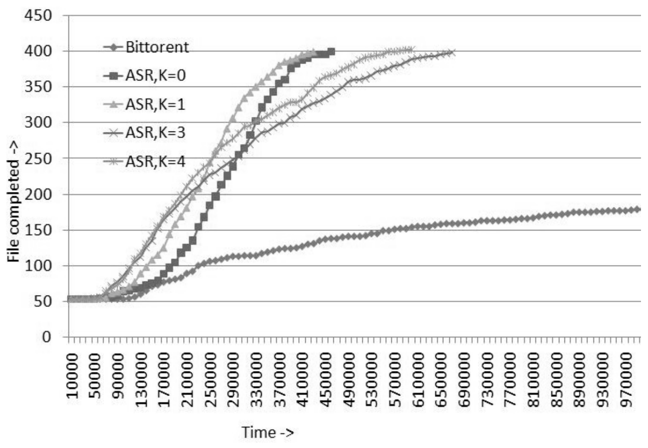

Fig. 2 Download Completed

Note that the distribution of the shared file for the other nodes participating in the network (namely the nodes that are neither seeder nor new) is picked uniformly at random between 10% and 90%. All the tests has been made using the UniformRandomTransport layer.

Above Fig.2. shows the downloading times achieved by ASR with different values of K. It shows that ASR finished downloading for first 350 peers much faster than Bittorent without ASR. It is also clearly showing that the k values have a affect on over all downloading rate , with highest downloading rate when k=1 i.e when one closest peer is among the nodes which are unchoked .

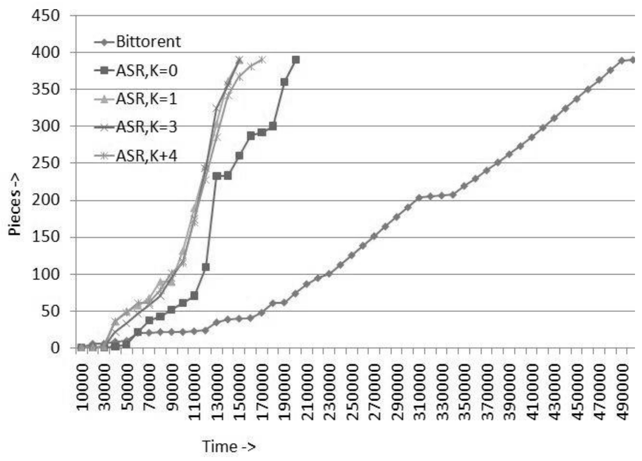

Fig. 3 Download Rate

Next , we study the affect of various K values for individual downloading rate for pieces. Fig. 3 shows the outcome ,where K > 0 leads to high personal piece downloading rate for leechers.

The result of modified unchoking in above figures are under the varying K values.

Fig. 4 shows loss in the search radius for leechers with different k values used in unchoking algorithm. Clearly there is a huge gap in the hops when compared with normal bitTorrent . Which shows loss in trans network traffic as peer weather leechers or seeders both have taken in account locality as important factor while processing there peer selection algorithm.

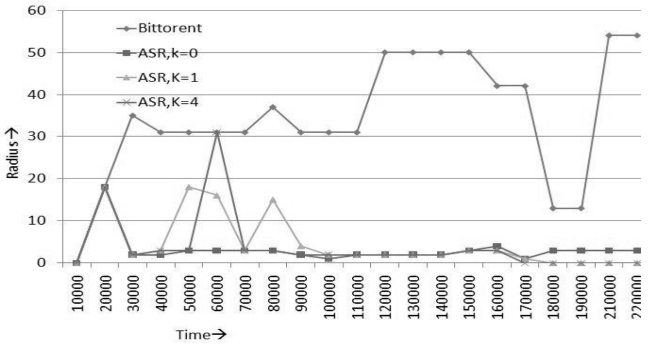

Fig. 4 Search Radius

Main reason for choosing peers with high upload rate is to avoid free riding as it will encourage more and more peers to upload data on the network. So we compared what is the upload rate in terms of block /sec by different seeders for normal BitTorrent as well as for Different K values. Fig. 5 shows the upload rate graph ,clearly depicting that upload rate by using ASR with K=0 is slightly higher than normal BitTorrent. It also shows that with higher K values i.e with K= 2,3,4 the upload rate is very unstable with high at the beginning and low at the end. Unchoking peers N-K peers with K= 0,1 is showing moderate upload rate for the seeders.

The graphs shows that for K>=1, the download rate do not change much ,so K=1 should be used as it provides best upload rate for the seeders.

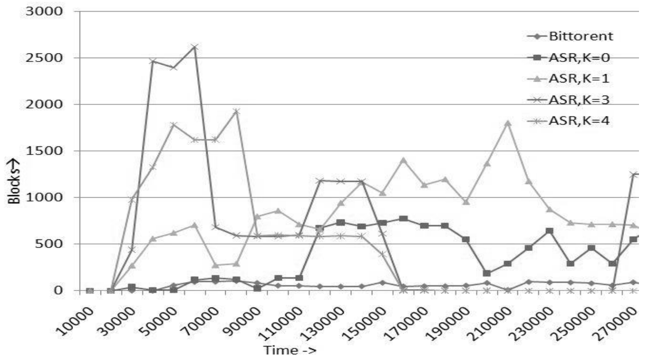

Fig. 5 Upload Rate

V. Conclusion And Future work

This paper presents a analysis of experiment trying to have a fare balance between trans network traffic problem and free riding problem. The changes suggested in this paper for unchoking algorithm of BitTorrent do not require any overall change in the algorithm and provides better uploading rate to the seeders as it is the main goal of the seeders to have high upload rate as this will help them to have higher priority over other peers while downloading.

There a lot of variable things which need to be looked into while making a peer selection algorithm e.g. bandwidth. Locality, availability of pieces, frequency of peer on the network etc . In the future , we intend to study and propose a peer selection algorithm which take all these factors into account and suggest best sequence of peers, that a peer can use to communicate in order to optimize the efficiency of p2p file sharing protocol.

References:

[1]. IPOQUE, "Internet Study 2007: The Impact of P2P File Sharing, Voice over IP, Skype, Joost, Instant messaging, One -Click Hosting and Media Streaming such as YouTube on the Internet," www.ipoque.com/resources/internetstudies/internet-study-2007, February 16, 2009.

[2]. R. Bindal, P. Cao, W. Chan, J. Medval, G. Suwala, T. Bates, A. Zhangan," Improving traffic locality in BitTorrent via biased neighbor selection," in: Proceedings of the 26th International Conference on Distributed Computing Systems (ICDCS 2006), Lisboa, Portugal, 2006.

[3]. S. Ren, E. Tan, T. Luo, L. Guo, S. Chen, X. Zhang, "TopBT: a topology aware and infrastructure-independent BitTorrent client," in: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM 2010), San Diego, USA, 2010.

[4]. L. Sheng, H. Wen, "Nearby neighbor selection in p2p systems to localize traffic," in: Proceedings of the Fourth International Conference on Internet and Web Applications and Services (ICIW 2009), Venice/Mestre, Italy, 2009.

[5]. Wei LI, Shanzhi CHEN, Tao YU, "UTAPS: An Underlying Topology-aware Peer Selection Algorithm in BitTorrent," in: Proceedings of 22nd International Conference on Advanced Information Networking and Applications,(ICAINA 2008)Japan,2008.

[6]. Bi Liu, yanchauan Cao, Yi Cui, Yansheng Lu, Yuan Xue," Locality analysis of BiTorrent Like peer to peer systems",in : Proceedings of IEEE CCNC 2010.

[7]. D.R. Choffnes, F.E. Bustamante, "Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems," in: Proceedings of the 7th IEEE conference on Consumer communications and networking conference .Las vegas,Nevada,USA.Pages 1208-1212

[8]. Fenglin Qiny, Ju Liuy, Lina Zhengy and Liansheng Ge," An Effective Network-Aware Peer Selection Algorithm in BitTorrent",in: Proceedings of Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009

[9] R.L. Pereira, T. Vazão, "On the cohabitation of Adaptive Search Radius enabled peers with regular eMule peers," in: Proceedings of the IEEE International Conference on Communications (ICC 2007), Glasgow, Scotland, 2007.

[10]. <http://www.bittorrent.org/beps/bep_0001.html>

[11]. Ricardo Lopes Pereira, Teresa Vazao , Rodrigo Rodrigues, "Adaptive Search Radius – Using hop count to reduce P2P traffic", Computer Networks 56(2012).

[12] <. http://peersim.sourceforge.net/>