

# ISP Friendly Peer Selection in BitTorrent

Tharidu Fernando<sup>#1</sup>, Chamath Keppetiyagama<sup>#2</sup>

<sup>#</sup>University of Colombo School of Computing  
No 35, Reid Avenue, Colombo 7, Sri Lanka

<sup>1</sup>tharidu@std.ucsc.cmb.ac.lk

<sup>2</sup>chamath@ucsc.cmb.ac.lk

**Abstract**— Peer-to-peer (P2P) applications provides services such as content sharing, video-on-demand, voice-over-IP which are very popular among Internet users and it accounts for a significant amount of global internet traffic. Use of overlay networks pose significant new challenges to Internet Service Providers (ISP). In order to reduce operational costs ISP are throttling or blocking P2P traffic which is unfavourable for a majority of internet users. In this paper, we design and evaluate an approach of reducing these costs when using BitTorrent P2P applications. Our approach involves of using knowledge of network paths that can be gathered in trackers and using it to bias the peer selection without any modification to any client software. Using the results of an evaluation of nearly 300 users in PlanetLab testbed the proposed approach will reduce cross-ISP traffic by 44% while improving in download speed (40%) and upload speed (39%).

**Keywords**— Peer-to-peer, BitTorrent, traffic locality, performance, neighbour selection

## I. INTRODUCTION

Peer-to-peer networking is a distributed application architecture which is built at application layer. Today various middleware such as Network Address Translation (NAT) and firewalls destroy the end-to-end nature of the Internet, however P2P complements the end-to-end nature. Nowadays P2P systems enable important scalable and reliable services such as video streaming and Voice over IP (VOIP).

P2P applications are self-scaling, therefore they are able to cope up with the supply. Compared to other network technologies which have limited bandwidth in a P2P application the peers participating provide bi-sectional bandwidth which utilizes the existing resources available more efficiently.

A recent study [1] conducted in Asia-Pacific region in 2011 found that during a peak traffic period over 40% of the traffic is P2P applications such as peer casting and BitTorrent. Over 40%-70% of Internet traffic worldwide is P2P applications [2] which shows the dominance over traditional client-server network technologies and the fixed infrastructure content-distribution networks.

Current P2P implementations are designed without any regard to existing Internet topologies. This have led Internet Service Provider (ISP) to a no-win situation.

Since users demand more bandwidth for P2P application they have found to be upgrading their Internet connection which results more revenue to ISP [3]. On the other hand they suffer significant link costs because of the P2P traffic is swamping the ISP bandwidth and they generate a significant amount of cross-ISP traffic.

This means that P2P traffic will often crosses multiple network boundaries which is not good for ISPs because most network bottlenecks in the Internet are assumed to be either in access networks or in links between the ISPs [4]. Therefore P2P traffic have increased bandwidth costs of ISP which has

driven to throttle or block P2P traffic which will be unfavourable towards a larger percentage of the Internet users who consumes P2P applications.

However ISPs are unable to block or throttle P2P traffic because the used ports can be easily changed or the traffic can be encrypted. Early on, port blocking was effective but nowadays it is an ineffective way therefore ISPs' are using traffic shaping devices which resides near the edge routers of ISPs. However the traffic shaping done by ISPs' reduce the bandwidth for P2P applications but do not try to improve it using the available bandwidth. Some ISPs are placing caches at the ISP gateways to reduce P2P traffic [5] which is questionable and raises legal issues since ISPs may participate in distribution of illegal copyrighted material which may cause severe legal action.

The origin of the problem arises from routing in overlays that exists in P2P systems. In the Internet an Autonomous System (AS) is a collection routing prefixes under several network operators which define the routing policy to the Internet. The path is usually the shortest path routing based on a fixed per-link cost [6]. However, P2P systems discard the available routing but rather implement own routing methodologies which aggravates the problem.

A popular P2P application such as BitTorrent enables peers to share content over networks. In BitTorrent a tracker is used to distribute the peer information which uses a random algorithm to provide peers who have the required content [7]. The random peer selection is the root cause of generating a significant amount of cross-ISP traffic.

Although proximity of the peers can be taken into account when selecting the peers because it can be assumed if the peers are near, there is a high probability they can be in the same ISP. However this is not currently implemented in BitTorrent.

A majority of Internet consumers participate in P2P applications such as BitTorrent, however shortcomings of the application itself and restrictions imposed by ISP makes it hard to use. Since BitTorrent represents a major proportion of the network traffic today, rectifying its shortcomings and optimizing will lead to more efficient BitTorrent applications without congesting ISP bandwidth.

As observed alternative approaches ([6], [8], [9], [10]) which tries to solve this problem but cannot be deployed easily because in order to work, majority of software will have to be extended or modified. In this paper, we expect to use similarities that exist in peers to find a better alternative and a scalable approach on peer selection without breaking the available existing infrastructure. The technique is based on AS hierarchy that exists in Internet which connects ISPs. Some ISPs connect directly to others without using the bandwidth of higher tier ISPs. Therefore based on the encountered routers to a specific two Internet Protocols from one location it is possible select a more optimized IP with a lower network

distance path. This approach does not require any cooperation between ISPs and subscribers while incurring a minimum overhead. In this paper we prove this to be very successful, demonstrating that minimal topology information can greatly enhance the peer selection algorithm that reduces cross-ISP traffic.

Using a test environment setup in PlanetLab we were able to achieve a 44% average AS hops reduction. The algorithm was selecting 13% more peers that are 2 hops or less which greatly reduces cross-ISP traffic. Download and Upload speeds were increased by 40% and 39% respectively which improves the transfer performance significantly which will be user perceivable.

This work provides the following significant contributions:

- A description of our novel biased peer selection algorithm which gathers topology information at low cost without changing any of the available infrastructure except tracker.
- An implementation of the proposed approach that has been deployed in PlanetLab and evaluated using nearly 300 nodes.
- An analysis of the measurements using statistical techniques proving that cross-ISP traffic can be significantly decreased. Our technique not only reduced cross-ISP traffic but improved download and upload speeds also.

The rest of the paper is organized as follows. In section II we reviewed the background information relevant to our work. In section III we provide a high level design of our approach. Section IV discusses the implementation details of the proposed approach. In section V we present a detailed evaluation and interpretations of our approach. Section VI concludes the paper and section VII discusses possible future work of this work.

## II. BACKGROUND

Most P2P systems that exists use an arbitrary peer selection algorithm which ignores underlying Internet topology and costs occurred by ISP. These systems connects to a randomly chosen endpoints crossing multiple ISP networks while it is possible to get the same content by nearby peers. BitTorrent is one of the most popular P2P application because of its scalability and relatively high performance compared to other P2P applications.

### A. BitTorrent

BitTorrent is a P2P file sharing service designed to distribute large files over a vast user community [11]. As of January 2012 BitTorrent user community exceeds 150 million. It uses tit-for-tat mechanism as a method of seeking pareto efficiency [7] and achieves the scalability and efficiency by making use of the bilateral bandwidth available in the participating nodes or peers. It encourages reciprocation by giving more priority to nodes who upload more data and discourages less priority to nodes with less uploads speeds.

### B. File Distribution

To initiate a BitTorrent deployment a user is required to create a static meta-info file called "torrent" file which includes the SHA1 hashes of 256 Kilobytes blocks of the files [11]. The static torrent files are hosted in a torrent repository.

Trackers implement a simple protocol built on top of Hypertext Transfer Protocol (HTTP) which helps peers to find out each other. The existence of the trackers makes BitTorrent not completely centralized but a hybrid. In BitTorrent a seeder is a node who had completely downloaded the distributed file (or files). Seeder must be alive at least till he uploads a copy of the complete file to the other nodes.

Leecher is a node who has zero or partial content of the files. Swarm is all the peers (including the seeders) who shares a torrent. The connections between the peers be operated over Transmission Control Protocol (TCP) and use the PcrWire protocol [11].

1) *Peer Selection*: When a peer is interested in downloading a file he must first download the torrent file and start it in the client software. Then the peer will be connected to one or more trackers which will provide the initial peer list of nodes who are downloading the same file.

The tracker's responsibility is to keep simple statistics about the torrents and provide a random subset of the peers to the peers. Since all logistic communication is handled directly through peers so tracker does not need a large bandwidth to operate.

In a defined interval peers will announce the owned blocks by them to other peers resulting a low bandwidth overhead. Since peer selection is done randomly it follows a power law graph [11] which brings it robustness properties. However this randomness has caused major problems to ISPs which will discuss later.

In later iterations of BitTorrent it supports Distributed Hash Table (DHT) deployment. In DHT peers will be able to get information about others without querying a tracker. Timpanaro et al. [12] recently showed that DHT approach is much more vulnerable than a tracker approach because of the lack of security infrastructure in DHT.

### C. Existing optimizations to BitTorrent

1) *Peer and ISP collaboration*: A number of researches is biased towards an ISP based solution. Aggrawal et al.[6] and Bindal et al.[9] argues that most preferable way is to use ISP as an oracle to provide a peer selection that is more limited to the highly connected ISP cluster and a limited visibility to the outer world.

Bindal et al. argues that the best ways is to implement the logic in P2P traffic shaping devices thus no need of changing client or tracker software. Aggrawal et al. has suggested to use a third-party service which will be hosted at ISP that provides other P2P nodes to get geographic information easily which can be eventually used to bias the peer selection.

These oracles do not recommend peers for performance improvement but to bias peer selection which will reduce the ISPs costs. The P4P project is also rather similar to above goal but it uses a framework which includes custom trackers for

both the ISP and P2P systems [13]. All these deployments ultimately will need the "trust relationship" between peers and ISPs in order to work.

2) *Peer and Content Delivery Network collaboration:* Choffnes and Bustamante suggest an approach that includes a third party Content Delivery Network (CDN) as an external oracle service needed to get biased peer selection [8]. Since CDN does not depend on the cooperation between the ISP and the peers this proved to be a more successful solution compared to other approaches. However users have to download and install a version of Azureus BitTorrent client and a plugin to participate in the network.

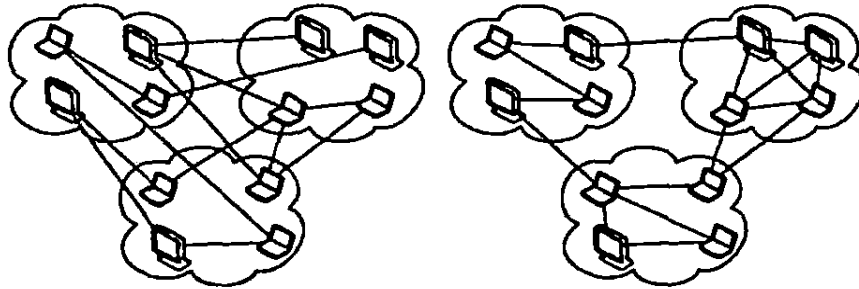


Fig. 1 Random peer selection (left) vs. biased peer selection (right)

Figure 1 shows a random peer selection and a biased peer selection. The cloud is an ISP and the right side figure which is biased peer selection is a result of the optimization. In biased peer selection there is a less number of connections to the nodes in other ISPs but more connections to nodes in same ISP.

This approach relies on the hypothesis of peers closer to the same CDN must be in the same ISP which proved to be accurate when analyzing the evaluation results. As above stated since every user had to install the client in order to use, this changes the infrastructure substantially and it limits the deployment capability.

#### D. ISP Costs

Peer-to-peer applications like BitTorrent is significantly different from traditional client-server applications because of the extensive use of upload bandwidth available. Regardless of the throughput, amount of data uploaded is equal to the amount of data downloaded therefore for a healthy BitTorrent network it is vital to have a considerable amount of seeders.

In normal web browsing user sends a small size request to the web server and it sends a large size reply to the user. This is favorable to the ISPs comparing the costs of download and upload bandwidths. However, in BitTorrent the upload bandwidth will be utilized completely occurring a large amount of upload data. This will directly hurt the ISPs because they charge the users a flat-charge normally.

As a solution ISPs are throttling BitTorrent traffic in order to reduce the utilization however this action is impractical because of the encryption features of BitTorrent. Some ISPs are moving into volume based packages which limits the upload and download capacity.

Two arrangements that allow networks to interconnect directly or indirectly are called transit and peering. It is important to analyze these two methods since it directly affects the ISP costs when BitTorrent is being used [14].

1) *Transit:* In this arrangement one autonomous network agrees to carry the traffic between two or more networks. A "transit fee" is charged for the service provided to other networks who carry traffic using the transit network and since a single network do not connect to all the networks, the transit provider will deliver some of traffic through his transit networks indirectly.

2) *Peering:* In peering two autonomous networks will agree to exchange traffic between them without any charge. This is more economical than transit however there will be a fixed costs establishing the peering setup which include expensive hardware.

However in the long run peering is much cheaper than transit however visibility of other networks is much lower. Therefore an ISP is required to have a transit network in order to connect to most of the networks. If we apply P2P scenario, ISP is much benefited if the traffic stays in peering network, or ISP network. But it is inevitable to use transit traffic in such applications. Therefore ISPs are occurring significant overhead charges as transit costs.

Although peering means there is a direct connection between two networks which will result in lower latency and lower loss of packets [15], this will rarely benefit BitTorrent because of the randomness of the peers. Most BitTorrent peers will be needing transit networks in order to reach other. As a result selecting peers who are in peering networks or in same ISP network will bring significant cost savings to ISP while giving a user more pleasant experience in terms of bandwidth and connection stability.

### III. DESIGN

#### A. Overview

To reduce ISP cross overs happening in BitTorrent we aim to bias it to peers with shorter network distances which inherently have smaller number of ISP cross overs. Achieving such feat can be done by modifying BitTorrent clients, however currently they are more than a dozen BitTorrent clients available in the Internet therefore it is not possible to deploy at a large scale easily.

A more probable method is to bias peer selection in the tracker level which will be transparent to the BitTorrent clients connecting. Although there are several trackers available, to use it is must easier to deploy to a larger number of BitTorrent users. To bias the peer selection the tracker must be aware of the connecting peers' network conditions. As earlier discussed in section II-C, one option is to use an ISP-based oracle. However deployment of such service will hurt the privacy of the users. We propose to use multiple existing third party services to bias the peer selection in the tracker.

#### B. Architecture

A widely available whois servers are used as a third party service which returns the AS number and the IP block of the sent IP address. Traceroute statistics to the peer will be compared with other peers' statistics to gather nearby peers.

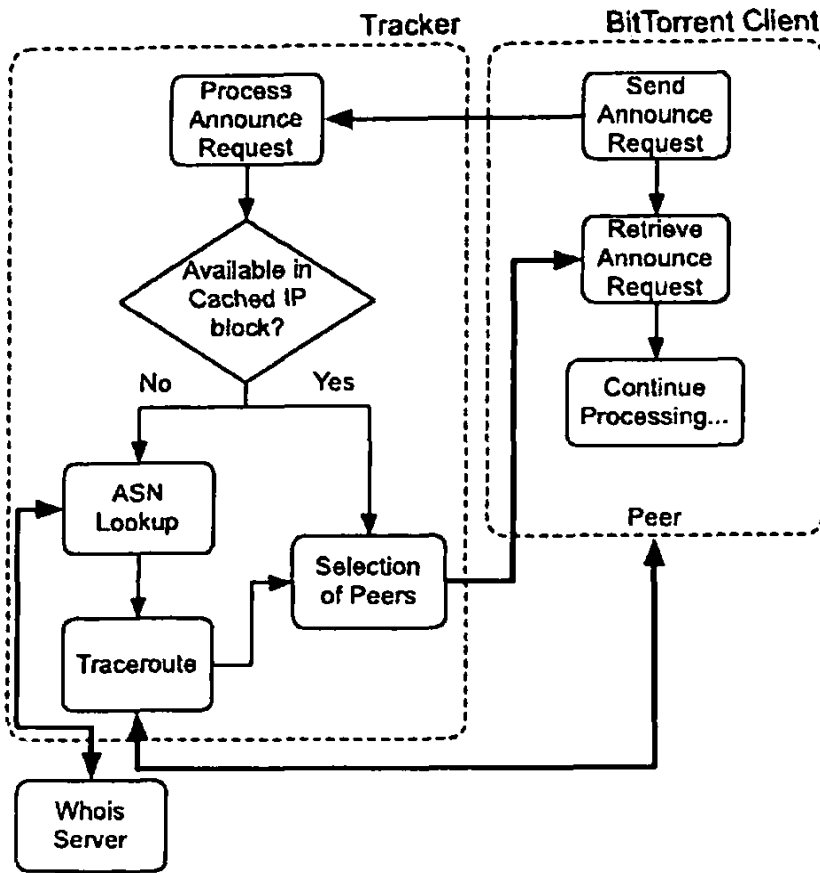


Fig. 2 Functionality of the tracker

However when evaluating on PlanetLab the BitTorrent client will also be modified to gather statistics of the quality of the peers connected such as trace routes and pings. The gathered data will be used to prove the hypothesis.

C. Design Components

1) *Autonomous Number Lookup:* Autonomous number lookup is facilitated by a widely available whois servers around the globe. It is done using the dig command that is provided by operating systems. Since in dig a Domain Name Server (DNS) lookup is executed it is cacheable by the system because the lookup is User Datagram Protocol (UDP). The lookup takes few milliseconds therefore the overhead is also minimum. Autonomous System Number (ASN) lookup can gather important statistics of the peer that can be used to locate optimal peers.

2) *Traceroute Statistics:* When a traceroute request is issued to a particular IP address it returns the intermediary routers or nodes in addition of latency and AS number.

Normal Internet consumers get Internet access from tier 3 ISPs while they get Internet access from tier 2 ISPs. Therefore some peers although do not belong to same ISP they can get internet from same tier 2 ISP. This means they do have less network distance between them so less number of cross overs are happening. Additionally there are Internet Exchange Points (IXPs) which reduces the use of upstream ISPs to exchange traffic. Thus we can use traceroute statistics to identify these same networks so we can select optimal peers from them although they are not in same ISP.

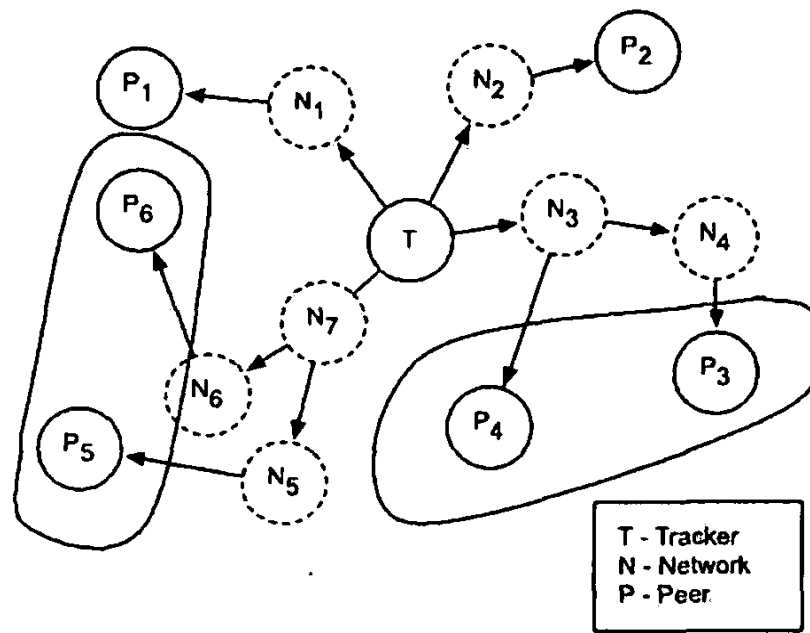


Fig. 3 Nearby peers using traceroute

In figure 3 the circled peers share a same network in the path to tracker. So P<sub>5</sub> and P<sub>3</sub> do not share any network between them so it will be less optimal to select them as bias peers to one another. If we chose above mentioned peers they will have a relatively large number of networks to cross over when communicating.

3) *Selection of Optimal Peers:* Tracker will do measurements to each peer it connects and stores information in the tracker database. When a peer request a list of peers for a torrent this database will be queried and the peers listed will be sent to the requesting peer. The peers will be selected according to the given below criteria.

- 1) Peers in the same ISP
- 2) Peers that share intermediary routers
- 3) Remainder is random peers

First criteria is checked by matching AS numbers. However at the initial stage of a torrent network there will not be peers in the same ISP. So using traceroute statistics we can get peers who share intermediary routes.

4) *Clustering of traceroute results:* Normally when running a traceroute against an IP address it prints all the intermediary nodes between the source and the destination along with the latencies recorded. If we analyse the latencies we can observe that there are two clusters visible. In figure 4 the clusters can be clearly observed. Mainly there are two clusters which are remote and nearby (local). The remote nodes are normally closer to the source while local nodes are closer to the destination. We will be using remote cluster nodes to get biased peers because they are the nodes which are in or close to ISP network where the peer resides.

Clustering is done using a basic cluster analysis called kmeans. Since our requirement is to divide the latencies to two clusters we use two as the k value in the algorithm.

Using this remote cluster we can identify routers which are near to the peer which are only few hops away. Local cluster represents routers which are close by to the tracker.

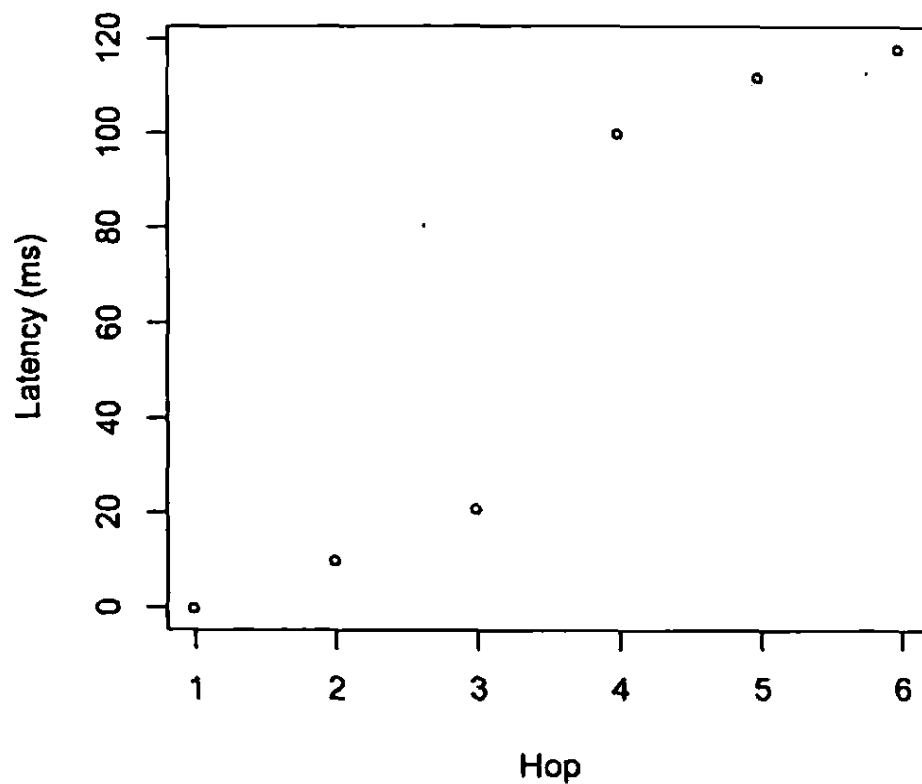


Fig. 4 Clustering of a traceroute result using latency

In algorithm 1 the overall pseudocode for selecting the peers is presented. Initially it lookups the AS number of the requesting peer. Then it compares to other AS numbers of the peers previously listed to the specified torrent. If there are peers with matching AS numbers that peers will be selected as biased peers.

Then it will traceroute the peer and cluster the results using 2-means algorithm. After that it will compare each and every AS number in the remote cluster to the previously listed peers. The matching peer's traceroute results will be then again clustered using 2-means algorithm. If the matched AS number occurs in both remote clusters of each peer the peer selected from the database will be listed as a biased peer. Finally, to not to select the same peer list for several peers every time it requests some peers will be randomly selected.

#### IV. IMPLEMENTATION

In this phase we turn the design presented in the previous chapter in to working code. According to the main components identified by chapter III we discuss the implementation of these components in this chapter.

Since the target of this research is to prove the hypothesis, instead of coding the required components from scratch we altered existing open source applications accordingly. All the implementation will be done at tracker so we chose a popular PHP tracker called BitStorm which provides it functionality in a single file. We used it with MySQL server as the database which keep track of required data and scales up the tracker. Since this tracker runs on a web server it will cater any torrent client regardless of the client platform.

Although the proposed design do not need to alter the client program, in order to show that biased peers are relatively better than the random peers we alter it. In a real world implementation this will not be required. As the client software we chose ctorrent which is a command line C++ program that can be run in majority of Oses. We chose a command line client mainly because for evaluation purposes

we had Virtual Machines (VMs) running a version of Fedora Operating System (OS). Therefore we can run the torrent client in the VM's terminal.

---

#### Algorithm 1 Bias peer selection

---

```

1: function GETBIASEDPEERS(peerIP)
2:   ASN ← lookupASN(peerIP)
3:   for each peer in peerDatabase do
4:     if ASN == peer.ASN then
5:       Add peer to BiasList
6:     end if
7:   end for
8:   traceroute ← traceroute(peerIP)
9:   cluster ← GetRouterClusters(traceroute)
10:  for each remoteCluster in cluster do
11:    remoteASN ← remoteCluster.ASN
12:    records ← Get records matching remoteASN DB
13:    for each peer in records do
14:      peerCluster ← GetRouterClusters(peer)
15:      if ASN in remote cluster of peerCluster then
16:        Add peer to BiasList
17:      end if
18:    end for
19:  end for
20:  if count of peers in bias list is not enough then
21:    Randomly select from the remaining peers
22:  end if
23: end function

```

---

#### A. Tracker

1) *ASN Lookup Component*: To query the AS number we used the dig command that is available in \*nix systems. To lookup the AS number we have to use the peer IP address. Out of several hundred WHOIS services we used the Team Cymru server for our implementation. It provides us with the ASN number and IP block that the queried IP address resides when it is supplied with +short command.

2) *Ping Component*: Latency between the tracker and the peer is important when selecting peers. In order to implement this we used ping command that exists in \*nix systems.

To reduce the performance overhead we do not resolve any IP addresses and we are averaging results of five pings to remove the fluctuations that can happen during the measurement. Finally the average round trip time will be stored. However this component may not work sometimes to some peers because of the network security. Since ping normally works using Internet Control Message Protocol (ICMP) packets peers who are behind a firewall which rejects ICMP packets may not respond such requests. Although the effect of this will be minimized using the traceroute component which uses TCP.

3) *Traceroute Component*: Collection of traceroute measurements is implemented using the traceroute command available in \*nix systems. In this component also we do not resolve any IP addresses but performs ASN lookups on any intermediary nodes. The results will be parsed and stored in

the database which will be used when selecting peers to other peers. Arguments will be provided to the command in order to resolve AS numbers, not to resolve host name and limit each node to 5 seconds of waiting time. This will make sure the results will be quickly delivered with minimum network overhead.

#### B. Client

The torrent client program was modified to log below parameters.

- 1) Download speed
- 2) Upload speed
- 3) Download time
- 4) Connected peer IP addresses

Above mentioned statistics will be recorded to a log during the program execution. This log will be used to evaluate the quality of the biased peers.

### V. EVALUATION

Evaluation of distributed system like BitTorrent is very challenging because we cannot exactly replicate a real BitTorrent network repeatedly for experiment purposes. Furthermore for a strong proof it is a requirement to test the proposed BitTorrent optimization in a large network consisting of more than 100 peers because in real BitTorrent networks the numbers of peers are quite large. We evaluate the proposed implementation in a testbed created on PlanetLab.

Since in real BitTorrent networks initially there are only few seeders available but more leechers are trying to download. This can be recognized as a flash crowd scenario. This will also be a worst case scenario because once the network becomes stable with multiple seeders the biased tracker will be performing better because of the multiple seeders available.

We replicated this scenario on PlanetLab using an initially available high bandwidth seeder which will be available throughout the experiment. The leechers will be joining the network every 10 seconds so it will replicate a growing network therefore trackers will not have information about all peers initially. A set of peers will be leaving the network after a specified time duration after seeding which is a normal situation in real BitTorrent networks.

#### A. Dataset

Dataset collection was an important task of this research. Using the modified torrent client for every 5 seconds we measure the average download and upload speed. Furthermore it records the total elapsed download time and the IP addresses of connected peers.

After the completion of the experiment each node performs an ICMP ping to every peer the client connected and it records the average round time trip of 5 ping tests. Finally it will perform traceroute to each peer the client connected. Traceroutes provide a view of the path between the hosts giving intermediary routers. Although one ISP may have more than one router, we are focusing on AS numbers which will

get a good representative of the cross-ISP links. The AS mappings are retrieved by a DNS service.

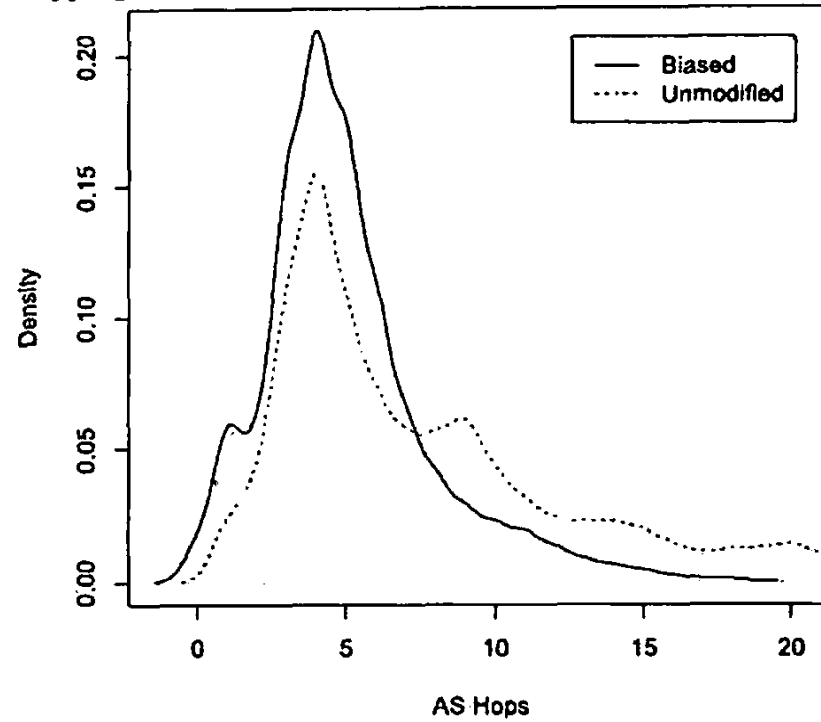


Fig. 5 Distribution of average number of AS hops

#### B. Cross-ISP Traffic

The main goal of this research is to decrease cross-ISP traffic occurring in BitTorrent networks. In this section we try to investigate whether the modified tracker reduced cross-ISP traffic.

Figure 5 presents the density curve of number of AS hops observed during the experiment. It is immediately clear that unmodified tracker occurs more AS hops compared to the biased tracker. The mean AS hops of unmodified tracker is 9 but in biased tracker it is only 5 hops which is a significant improvement of 44%. If we analyze furthermore the biased tracker selected peers who are only 2 hops away or less, 13% of whole dataset compared only 6% in unmodified tracker. Therefore bias tracker has more probability of selecting a closer peer in terms of AS hops.

#### C. Path Characteristics

We have clearly shown that reducing cross-ISP traffic is occurring significantly with the biased tracker. In figure 6 the distribution of ICMP ping Round Time Trip (RTT) is shown. The biased tracker clearly improves the density of lower latencies. This is a result of biasing tracker providing same ISP peers with < 5ms RTT. The median latency of biased tracker is 86ms while unbiased tracker's median latency is 126ms which is a considerable decrease (31% lower). Biased tracker achieves a considerable amount of peers lower than 10ms latency compared to unbiased tracker. Less latency means that peers will experience connections with more stability without occurring frequent loss of connection.

#### D. Transfer Performance

By observing the results of the biased tracker in cross-ISP traffic and path characteristics it can be assumed that average download and upload speeds will also benefit. In figure 7 the distribution average download speed is shown. If we analyze the distribution curve the biased peers have a faster average

download speed than unbiased peers (40% increase). The CDF curve also displays the same observation. Since this experiment is done on PlanetLab where all nodes have higher bandwidth links compared to average home users, the average download speed is much higher.

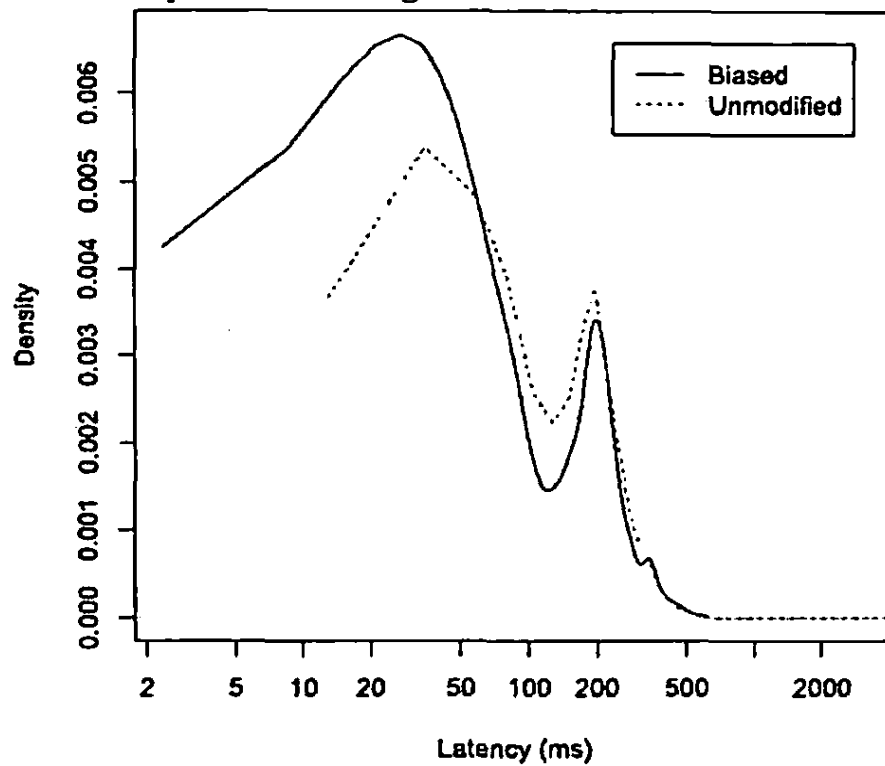


Fig. 6 Distribution of average ICMP ping round trip time

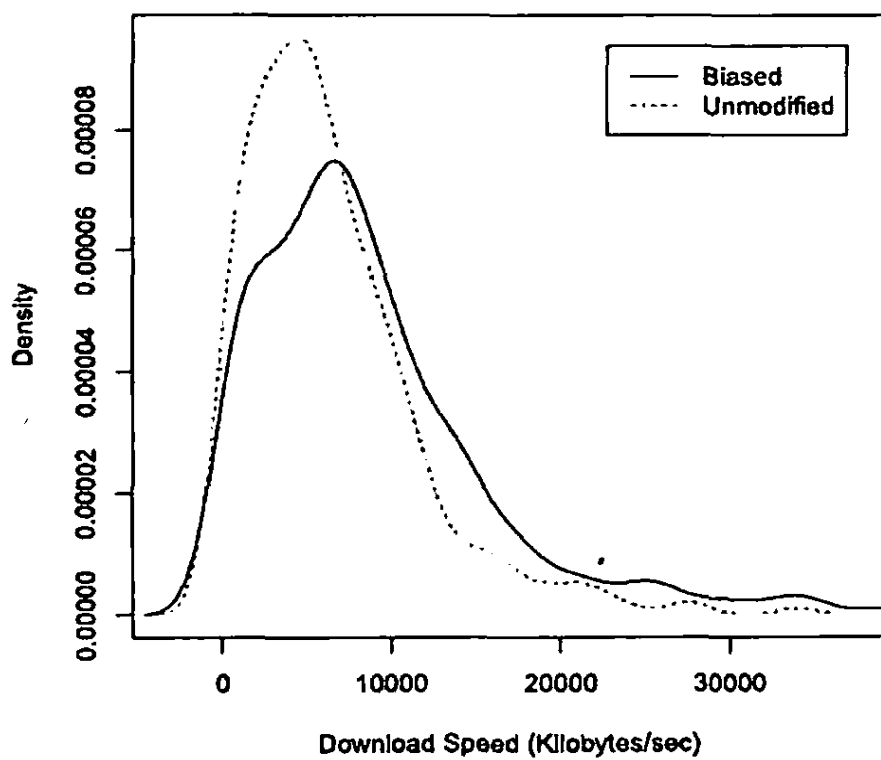


Fig. 7 Distribution of average download speed

In an actual deployment it is possible that biased peers will not occur any significant download speed increase due to bandwidth limitation of peers. This is because the current design of the biased tracker do not take account of the bandwidth available at a peer. Therefore it is possible some peers will be overloaded. In ISPs where there are higher upload bandwidth available the implemented biased tracker will perform much more better. However in ISPs with less upload bandwidth this effect will be reversed.

It is favorable for ISPs to change their bandwidth accordingly which will promote of connecting inter-ISP peers

which will increase the performance while reducing cross-ISP traffic.

If we analyze average upload speed of peers which is shown in figure 8, we can observe that biased peers have significantly higher upload speeds than the unbiased peers (39% increase). This is obvious because of the average download speed also has a similar increase. The use of traceroute and dig commands will incur an overhead at the tracker but this is insignificant compared to unbiased tracker. Therefore there will be a minimum overhead at tracker.

#### E. Multiple Seeders Network

In an environment where there are more than one seeder available, as we observed the biased tracker performed better because of the stability of the network and multiple amount of nodes available to select. A single seed network can be taken as a worst case scenario and this case exists rarely in real world. However this scenario can be observed at the beginning of the torrent network therefore it cannot be neglected.

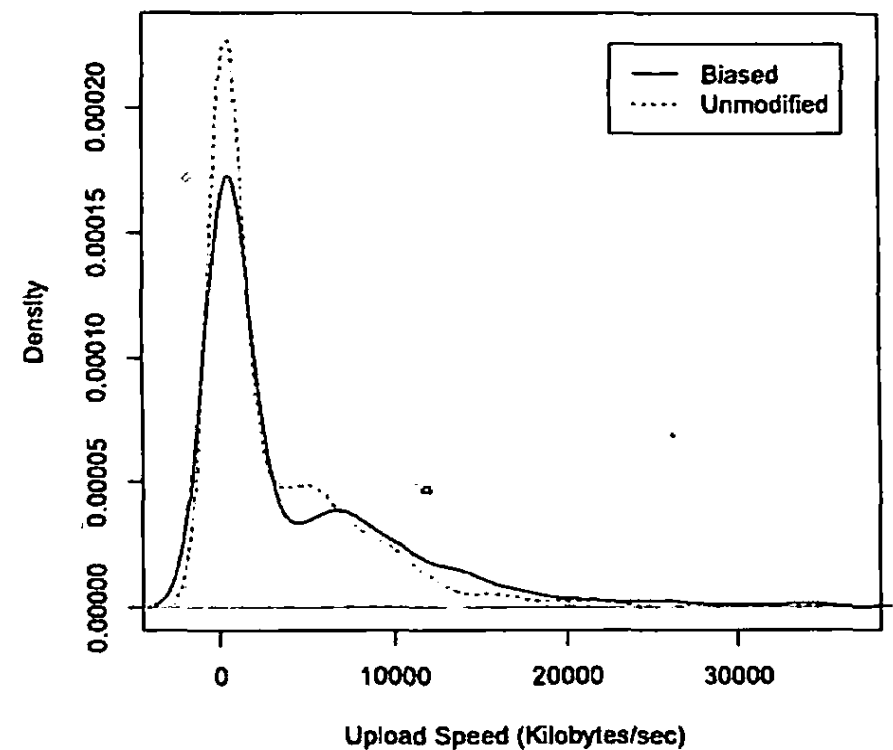


Fig. 8 Distribution of average upload speed

#### F. Statistical Significance

In this section we evaluate whether we can accept the proposed hypothesis by subjecting the dataset to a statistical significance test. In ideal experiment conditions we can pair two records from each dataset but due to network conditions it is not possible. The number of observations are also not the same in both datasets.

Therefore we selected a non-parametric test which is ideal because the test will not assume anything about the dataset which will lead to more accurate conclusions on the calculated p-values. We chose Mann-Whitney U test as the non-parametric test. Significance level ( $\alpha$ ) was set as 0.05 (normal value) and if the calculated p-value is less than it, we can reject the null hypothesis and accept the alternative hypothesis.

If we analyze the results of table I it is clear that AS Hops and latency is significantly improved because p-value is very

close to zero. Download and upload speed changes are also significant and well behind 0.05.

Therefore we can conclude that even though this research's main goal is to minimize cross-ISP traffic the proposed solution achieves that goal while improving on latency, download speed and upload speed.

TABLE I  
MANN-WHITNEY TEST RESULTS

Variable	P-Value	Significance
AS Hops	< 2.2e-16	Significant
Latency	< 2.2e-16	Significant
Download Speed	9.713e-06	Significant
Upload Speed	0.001886	Significant

Reduction of latency is evident because of less number of AS hops. Increase of download speed and upload speed will be favourable because the proposed solution utilizes the connected peer links bandwidth efficiently. Increase of upload speed is favourable for BitTorrent because it means more leechers are getting the file content quickly therefore availability of the files goes up. Finally according to this results we can reject the null hypothesis firmly.

## VI. CONCLUSION

It is clear that BitTorrent is a dominant part of P2P traffic that utilise the global Internet bandwidth. With the recent increase in involvement of ISPs to limit the bandwidth to BitTorrent in order to reduce heavy cross-ISP bandwidth charges it is critical to change the existing system. The researches that have been conducted in this research domain have significant advantages over traditional BitTorrent, but comes with a significant flaw, which is the difficulty of deployment in a large scale. Due to this problem these approaches will not be able to bring a quick solution to the discussed problem and to overcome it, it may take several years to disseminate with the right infrastructure (software).

Our approach can be deployed without harming the existing infrastructure in a significant manner. The existing tracker operators will be able to deploy the bias tracker with a few modifications and BitTorrent users do not need to change or upgrade their existing software. This can be furthermore improved if tracker operators choose to host a whois server or communicating with a geographically closer whois server. This will reduce the delay of collecting statistics of the peers. According to the results collected, in a large scale deployment it is clear that this approach reduce cross-ISP traffic while bringing increase of performance and stability to the users.

Although this study is constrained to BitTorrent this optimization will be also applicable to similar P2P applications with minimum changes and bringing the same benefits.

## VII. FUTURE WORK

As of future work we have identified several possibilities. One is to improve the selection algorithm to adapt into all stages of a BitTorrent network. This will ensure continuous improvement rather than only in later stages as of now.

Additionally we would like to explore the possibility of exploiting DHT to optimize peer selection. This is a high priority because of legal constraints some sites offer only a magnet link containing only the hash without any trackers.

Finally no testbed, no simulator, and no emulator is inherently representative of the Internet [16], therefore one of our major goal is to deploy this implementation in Internet because in larger networks how the biasing process will work is still to be explored.

## ACKNOWLEDGMENT

The authors would like to thank Mark Berman (GENI Program Office), Anjana Bandara (Statistics Department, University of Colombo), David Choffnes (University of Washington) for their support given.

## REFERENCES

- [1] Sandvine, "Sandvine global internet phenomena report: Asia pacific, fixed access, fall 2011," 2011.
- [2] H. Schulze and K. Mochalski, "Internet study 2008/2009," IPOQUE Report, 2009.
- [3] T.Mennecke. (2005) Dsl broadband providers perform balancing act. [Online]. Available: <http://www.slyck.com/news.php?story=973>
- [4] A. Akella, S. Seshan, and A. Shaikh, "An empirical evaluation of wide-area internet bottlenecks," in Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, ser. IMC '03. New York, NY, USA: ACM, 2003, pp. 101–114.
- [5] Sandvine. (2008) Peer-to-peer policy management. [Online]. Available: [http://www.sandvine.com/solutions/p2p\\_policy\\_mngmt.asp](http://www.sandvine.com/solutions/p2p_policy_mngmt.asp)
- [6] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can isps and p2p users cooperate for improved performance?" SIGCOMM Comput. Commun. Rev., vol. 37, pp. 29–40, July 2007.
- [7] B. Cohen, "Incentives build robustness in bittorrent," in Workshop on Economics of Peer-to-Peer systems, vol. 6, 2003, pp. 68–72.
- [8] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems," SIGCOMM Comput. Commun. Rev., vol. 38, pp. 363–374, August 2008.
- [9] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, "Improving traffic locality in bittorrent via biased neighbor selection," in Proceedings of the 26th IEEE International Conference on Distributed Computing Systems, ser. ICDCS '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 66–66. A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [10] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in bittorrent," in In NSDI'07, 2007.
- [11] B. Cohen, "The bittorrent protocol specification," [http://www.bittorrent.org/beps/bep\\_0003.html](http://www.bittorrent.org/beps/bep_0003.html), Jan. 2008.
- [12] J. Timpanaro, T. Cholez, I. Chrisment, and O. Festor, "Bittorrent's mainline dht security assessment," in New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on. IEEE, 2011, pp. 1–5.
- [13] H. Xie, A. Krishnamurthy, A. Silberschatz, and Y. Yang, "P4p: Explicit communications for cooperative control between p2p and network providers," P4P Working Group, Tech. Rep., May 2007.
- [14] V. Valancius, C. Lumezanu, N. Feamster, R. Johari, and V. V. Vazirani, "How many tiers?: pricing in the internet transit market," SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. 194–205, Aug. 2011.
- [15] W. Norton, "Internet service providers and peering," in Proceedings of NANOG, vol. 19, 2001, pp. 1–17.
- [16] S. Floyd and V. Paxson, "Difficulties in simulating the internet," IEEE/ACM Transactions on Networking (TON), vol. 9, no. 4, pp. 392–403, 2001.