International Journal of Computing and Network Technology
ISSN 2210-1519
Int. J. Com. Net. Teach. 3, No. 1 (Jan. 2015)

New Technique to Improve BitTorrent Performance Based on Application Layer Traffic Optimization

Nattee Pinthong[1] and Woraphon Lilakiatsakun[2]

[1]The Electrical Engineering Graduate Program, Faculty of Engineering, Mahanakorn University of Technology, Bangkok, Thailand
[2]Faculty of Information Science and Technology, Mahanakorn University of Technology, Bangkok, Thailand

Received: 2 Sept. 2014, Revised: 20 Nov., 2014, Accepted: 7 Dec. 2014, Published: 1 Jan. 2015

Abstract: Currently, BitTorrent is the most popular protocol that is widely used for P2P file sharing system. Nevertheless, BitTorrent protocol exposes some inefficient processes that are choosing neighbours and selecting pieces. These are mainly based on the knowledge of overlay topology without considering underlying internet topology that might cause of traffic problems such as the bottleneck at some point of network. Therefore, we propose the new approach for BitTorrent protocol with Tracker Localized Algorithm, Picker Localized Algorithm and Chocker Localized Algorithm which autonomous hops between peers are calculated to make efficient decisions. Then to compare the AS hop between 5 AS and 7 AS in the network. We conducted simulation of our scheme based on the PeerSim. The simulation results show that the peers provide better piece downloading on BitTorrentunderlying topology and interacting among peers more efficiently. The Tracker localized algorithm outperforms the original BitTorrent for reducing the average downloading time of 20%. The breaking time of peers is the least when using Chocker localized algorithm. Additionally, the results show that ours scheme can decrease the peer-to-peer traffic and optimize traffic distribution for the whole network.

Keywords: BitTorrent Performance, P2P, ALTO

1. Introduction

Peer-to-Peer (P2P) system has emerged as a successful architecture for content sharing over the Internet. One of the most successful protocol for P2P file sharing system is the BitTorrent which was introduced by Bram Cohen [1] in 2001. Since then, BitTorrent has attracted significant attention from software developers and researchers for its wide deployment and there are many P2P application based on BitTorrent protocols in the software markets. The key success of BitTorrent protocol is its high efficient downloading performance [2]. The high downloading rate is achieved by choosing 4 best peers to provide chunks of targeted files and then simultaneous downloading from those peers.

Recently, researcher has conducted the study of the effectiveness of BitTorrent systems [3],[4]. The most recent work [5] has showed the stability of BitTorrent systems through a fluid model, and verifies the effectiveness of its incentive mechanism. They showed that the success of BitTorrent comes mainly from the ability to distribute resource consumption among the participate entities and avoiding the bottlenecks of centralized distribution.

However, the high amount of traffic in BitTorrent system makes several network resources irrationally utilized and would make bandwidth of other internet applications such as web traffic starvation and networks finally become bottleneck. In fact, previous studies [6]-[8] have shown that peers possibly request for downloading with another peers outside their local networks even though the required chunks are locally available. This exposes an inefficient downloading performance. Furthermore, as known, BitTorrent algorithm uses rarest-first technique to select downloading chunks and tit-for-tat method to choose neighbors, such techniques might cause unreliable performance for the whole networks. Besides, the limited links may be occupied by the network traffic and these network would be overloaded.

Thus, we propose the new technique to cluster of peer-to-peer nodes within an autonomous system to help peers in the BitTorrent system by using efficient indexing resources and transmission of piece and route information. We finally conduct simulation-based performance evaluation of proposed techniques and original BitTorrent system. The results have also shown that our scheme outperforms to the original BitTorrent.

E-mail: nattee@ieee.org, woraphon@mut.ac.th

The remainder of paper is organized as follows. In Section II, we explain the BitTorrent and ALTO (Application Layer Traffic Optimization). Then, related works are described in section III and our proposed algorithms are presented in Section IV. In Section V, we evaluate the performance of our algorithms using event-driven simulation. Finally, section VI is the conclusion of the paper.

2. BACKGROUND

A. The BitTorrent Protocol

BitTorrent (BT) is a very scalable Peer-to-Peer protocol for large scale content-distribution over the Internet. The BitTorrent works by splits file into pieces, where users connected to each other directly to upload and download portions of a large file (called as a piece) from other peers who already have entire file or parts of it. File chunks can be delivered as non-sequential manner, the hashes of which are included in a torrent file. All of these torrent files are stored in a server. The BitTorrent differentiates between two types of peers: leeches and seeds. Leeches are peers that only have some or none of the data while seeds are peers that have all the data but stay in the system to let other peers download from them. Thus seeds only perform uploading while leeches download pieces that they do not have and upload pieces that they have. A tracker in BitTorrent traces participation peers in a swarm. The downloading process is as follows. Every peer that participates in sharing a file is member of a swarm, which is tracked by a tracker, and multiple swarms associated to a single file can coexist in parallel. The set of all swarms and thus all peers sharing the same file is referred to as a torrent. If a new peer wants to download a file, it first connects with the tracker to get a peer- list which has the file it wants to get. Then the downloader can connect with these peers and download pieces in parallel from the peer in peer list. If the downloader completes the download process it begins to upload the file to other peers for free.

BitTorrent implements a set of algorithms that balances the content distribution load among a swarm of peers and overlay mesh network of peers. Each swarm is managed by a centralized process a tracker. The tracker does not host any content but maintains metadata about it. As leechs enter the swarm they first connect to the tracker. The tracker returns a random list of peers that have the content. Each leech then randomly selects a subset of that list as its neighbors and initiates requests to set up connections with these neighbors. This mechanism provided faster download whereas in normal P2P systems, the peer downloads from a single peer alone and the download speed is limited. Instead of downloading directly from the server, each leech requests pieces from all the peer it is connected to. The Tit-for-Tat (TFT) policy [9] in which downloader give upload preference to peers that provides high download rates, thus creates an incentive for peers to upload their data to other. The downloader periodically updates the connected peers to fine better neighbors. The downloader users a Rarest First policy [10] to determine which piece to request next. The BitTorrent employs a Tit-for-Tat policy to deter free riding, when the peers behave selfishly and use the swarm only to download pieces without making any contribution to the swarm. The peers exchange data on a Tit-for-Tat policy in the swarm of peers interested in the same file, the neighbors of a peer that provide the most data are allowed to request piece in return (call as an unchoking). The Tit-for-Tat policy construct thus creates an inventive for peers to upload their data to others. Once a peer has completed to download of a file, it can continue to seed it by using its upload capacity to serve the file to other for free.

1) Strict Priority

Peers concentrate on downloading all the piece of a file, before requesting for another piece. Thus if a sub-piece is requested, then subsequent subpieces of the same piece will be requested preferentially so as to complete the download of the whole piece as soon as possible, as only complete pieces can be exchanged with others.

2) Endgame Mode

When a peer has received or has outstanding requests for all pieces, requests for the not yet received blocks are sent to all connected peers that have the pieces. For every received block, the peer sends cancel messages to the peers that were sent requests previously.

3) Random First Piece

An exception exists to the rarest first policy when a peer first joins a torrent. Since the peer does not have any pieces, so it would like to download a whole piece quickly so that it can get ready to reciprocate with the TFT algorithm. In this case, the new peer will not take the rarest piece. Instead, it will download the first piece randomly in order to make sure it can get a whole piece as soon as possible.

4) Rarest First

Peers generally prefer to download pieces that are the rarest among their neighbours. This strategy is known as the rarest first algorithm [11] which works as follows. Every peer maintains a list of the pieces that each of its neighbours contains i.e., the pieces that are available from the fewest number of peers. This list is updated every time a copy of a piece becomes available from its neighbours. It then creates a rarest-pieces set which is the list of those pieces that have the minimum number of copies among its neighbours. It then chooses the piece to download that is rarest among its neighbours. This increases the popularity of the peer itself, since it now has a rare piece to share with the rest of the swarm and increases the chance of a peer exchanging with its neighbours as it has pieces that others require. This also benefits the seeds by reducing their server burden, since they need to upload less copies of the same piece. Furthermore, the exit of a seed will not leave the remaining downloaders without the opportunity to exchange file pieces, because this strategy assures that all the pieces are quickly distributed to maximum of the leechers.

5) *Tit-for-tat*

The tit-for-tat strategy [12] is taken up by a peer to preferentially upload to its neighbor peers who provide it the optimum downloading rates. A peer exchanges file pieces with a fixed number of neighboring peers. A leecher preferentially uploads to the best neighbors who provide it the best downloading rate and chokes others. Every few seconds, the leecher reevaluates the downloading rate from all peers who are sending data to it. This mechanism is very important to encourage downloaders and punish free-riders, thus preventing leechers from downloading without uploading anything.

6) *Optimistic Unchoking*

New peers are occasionally unchoked in order to discover potentially better opportunities [13]. Peers are thus given the chance to acquire their first pieces. If a peer uses only the TFT mechanism, there will be no opportunity for discovering other peers that can provide higher uploading rate. This strategy is extremely useful for newly joined peers to get started.

7) *Upload Only*

A peer becomes a seed, once it finishes downloading the entire file. Seeds cannot select peers based on downloading rates, as seeds have nothing to download. The seeds prefer to upload to peers with better uploading rates.

8) *Anti-snubbing*

A peer may be choked by the peers it was earlier downloading from, thereby getting poor downloading rates. To address this problem, when a peer notices that some time has elapsed without getting a single piece from a neighboring peer, the leecher assumes it is ‘snubbed’ by that peer and does not upload to it any further through regular unchoke.

It has widely been accepted that BitTorrent protocol provides very high transferring rate for P2P file sharing system. However, originally The BitTorrent protocol has not yet been considered the impact of network-wide traffic load. Heavy traffic load might cause certain parts of networks become clogged [14]

B. *The Application Layer Traffic Optimization (ALTO)*

As known, P2P applications do not consider locality when choosing their neighbors on their P2P process. As a result, certain participating peers may experience for long delays and ISPs might suffer from a large amount of costly inter-ISP traffic. To cope with these problems, several solutions [15] have been proposed to perform P2P localization. For instance, peers can measure message delay to other peers themselves to infer network distance in order to choose which peer would be communicated with.

Recently, the ALTO is the technique to provide network information to P2P applications to achieve better peer selection. We define The Application Layer Traffic Optimization from IETF an API through topology and infrastructure is request by the application layer and deliver by the network layer. The ALTO try to caching and replicate the piece to optimization all network traffic by distribution paradigm efficiency and extend by dynamic mechanisms that locate and determine distance to another peers which optimize infrastructure resources utilization. For instance, ALTO need to locate the nearest copy of file or the closest instance of a service among several available resources. Each ALTO server maintains a my-Internet view and my-Internet view consists of 1) a set of network locations 2) ALTO Cost between each pair of network locations for a given ALTO Cost Type. A Source Group is a set of network locations that have similar ALTO Costs to other network locations. A Destination Group is a set of network locations that have similar ALTO Costs from other network locations. A Source/Destination Group may represent an IP Prefix, a point of presence (PoP), a type of customers (wireless, DSL), an AS, or a set of AS. To construct network topology by IGP boundary, BGP location dependent visibility, collect IGP from link-state and BGP database then take into account area/level and AS boundary. Peers establish connections between randomly chosen subsets of cooperating peers from around the world. The distributed hash tables (DHT) use greedy forwarding algorithms to calculate the destination and making decisions them. This leads to the ALTO problem and how to provide the topology of the underlying network in the same time that allowing the requesting node to use some information of effectively and reach the node on which the content resides. Thus, it would appear that Peer-to-Peer networks with their application layer routing strategies based on overlay topologies [16]. To solve the problem is to build network coordinate systems which embed the network topology to enable network distance estimations based on latency.

The figure 1. shows the use of ALTO on P2P overlays A) ALTO Engine collects routing database such as OSPF or BGP Policy B) P2P clients finds content and list of peers such as IP address C) P2P clients sends the ALTO request with the list of address to rank D) ALTO engine reply ranked list of IP addresses and E) ALTO engine receives request and rank IP addresses based on location.

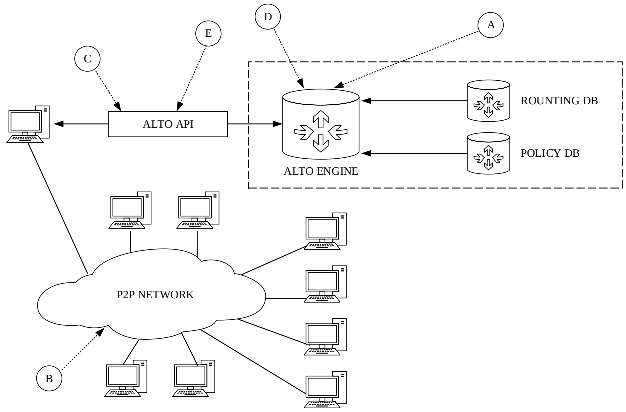


Figure 1. The use of ALTO on P2P Overlays

3. Related Works

Recently, many researches have been done to understand BitTorrent systems. The effectiveness of BitTorrent’s tit-for-tat and rarest-first mechanisms was considered in [17] and [18], the potential of network coding for content distribution was investigated in [19], and A. Legout et al. [20] to study from real experiments to advocate that the replacement of the rarest first and choke algorithms cannot be justified in the context of peer-to-peer file replication in the Internet.We instrumented a BitTorrent client and ran experiments on real torrents with different characteristics. The authors also showed experimental evaluation is peer oriented, instead of tracker oriented, which allows us to get detailed information on all exchanged messages and protocol events. We go beyond the mere observation of the good efficiency of both algorithms. Most of these efforts have focused on understanding the performance of single-torrent systems. Other works have analyzed the general characteristics of BitTorrent traffic and the impact of BitTorrent usage on the amount of inter-ISP traffic (e.g., [21]-[24]). Liu et al. [25] purpose the new scheme to improve the performance of BitTorrent by ALTO which make BitTorrent node aware of the topology of underlying networks and modify BitTorrent's original algorithms and replace them with three new localized algorithms based on autonomous system hops.There are a few recent works that consider multi-torrent environments. Neglia et al. [26] evaluated the benefits of multiple trackers in terms of improved tracker availability based on measurements. They found that multiple trackers significantly improve the availability and observed that multiple trackers can reduce the connectivity of the overlay. Guo et al. [27] study the limitations on torrent evolution in realistic environments, motivated by the analysis and modeling results. The authors also showed a graph based multi-torrent model to study inter-torrent collaboration, the model quantitatively provides strong motivation for inter-torrent collaboration instead of directly stimulating seeds to stay longer. Yang et al. [28] proposed rate-based incentives that motivate users to act as seeds for other torrents than they currently are downloading. Menasche et al. [29] purposed model to quantify content availability in swarming systems. The Author showed the model to analyze the availability and the performance implications of bundling, a strategy commonly adopted by many BitTorrent publishers today and find that even a limited amount of bundling exponentially reduces content unavailability. Perterson et al. [30] purposed the minimize download latencies for participants subject to bandwidth constraints and swarm dynamics based on a wire protocol that enables the Antfarm coordinator to gather information on swarm dynamics, detect misbehaving hosts, and direct the peers' allotment of upload bandwidth among multiple swarms.

4. Proposed Techniques

In natural of BitTorrent some algorithm such as choke/unchoke algorithm, peers select another peers whose download or upload rate is the fastest as their nearest peers in BitTorrent system but the peer selection algorithm are random. Then the method of peer selecting external networks is large. This probably leads to much unnecessary inter-ISP traffic. To overcome these problems, we introduce the creating of autonomous systems topology and the enhance algorithms in the following that to make the traffic of BitTorrent stay in the local network by using application layer traffic optimization approach, we try to adjust some methods in BitTorrent that can make to know some information of underlying topology. we obtain a peer’s AS number through its IP address and some public AS searching institutions. Then it’s not difficult to know which AS an IP address belongs to. After collecting the data, we know a map table of peers’ IP addresses and its corresponding with the AS numbers. Then we will show how to obtain an integrated AS topologic map through this map table. If all BitTorrent peers belong to N ASes, we select one peer randomly from every AS, then we run traceroute between any two peers to obtain their end-to-end route on IP layer. Next, we find all peers’ AS numbers in this route through AS-IP map table.

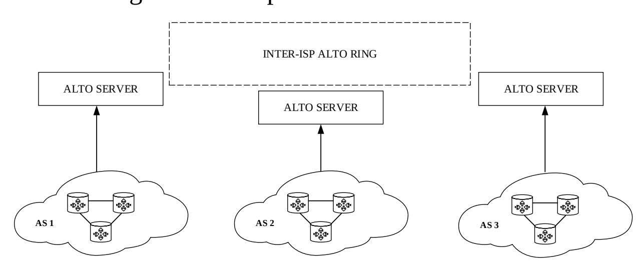


Figure 2. The inter-ISP ALTO Communications

Figure 2. shows The inter-ISP ALTO communications, the ALTO acquires routing information from within the AS then requests to receive within the AS are locally server, after that to requests received for addresses outside the AS that will be re-directed to

ALTO server locates in addresses' AS and then ALTO server exchange information.

A. Tracker Localized Algorithm

To follow up on the local characteristics of our algorithm to select peers.we receive a request from a peer by following the tracker to sorts by AS hop [31] of the candidate peer for the first peers and they will manage the tracking we assume that the initial number of peers in the candidate table are 128 candidates in the table if the number of peers is 128 or less than number of AS 128 the tracker send a request directly to the peer. If the number is greater than 128 and the number of ASis 128 is different from the 129then the tracker returns the first 128 peer for the first time if the number of AS 128 is the same as the number of AS 129 will be numberedof AS.

B. Picker Localized Algorithm

BitTorrent piece selection policy in effect, including strict priority, endgame mode, the rarest first and the random first. A rarest first algorithm is the factor of the traffic that occurs during ISP among these pieces selection policy. This deployment the picker localized algorithm replace of rarest first algorithm to encourage peer to download the first piece and the nearest piece. To downloads peer will count the distance to each piece. The distance is defined as the average number of AS jumping between peers who own this piece and downloaded, for example the 5 peers to own a piece to download peer, AS, they jump to peer downloads a 0, 2, 4, 8 and 16. Therefore, the distance of the 2 pieces to our algorithm is 2. The picker localized algorithm select pieces at a distance. The smallest value in the first download.

C. Chocker Localized Algorithm

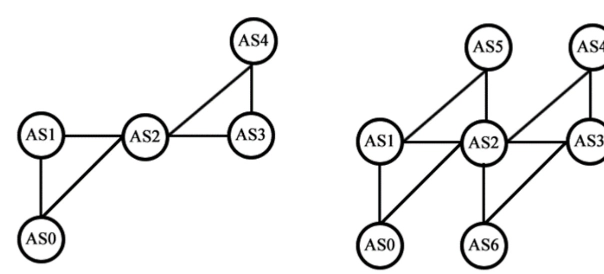
In BitTorrent the seeds select in 4 peers to download whose download rates of the largest to be unchoke and the download select of 4 peers to uploaded rates of the largest to be unchoke but the behavior is to ignore the position evaluation of peers and tend to make special traffic between ISPs. So we modified choking algorithm and unchoking algorithm to make peers unchoke their neighbors based on autonomous systems even distance of two peers rarely algorithms change the modification will not let peers tend to choose the same neighbors The reason is that choking algorithm and unchoking algorithm select appropriate peers, just in the interest of neighbors. The peers in neighboring will choose a piece collection is different from the peer requests. Although the collection of all the pieces to the continuous attention of a neighbor who is constantly changing with each point algorithm can ensure that the peer will exchange information with neighbors most of the effects.

We test the choking algorithm and unchoking algorithm policy similar in seeds help unchoking only the closest four neighbors who still has not finished downloading. But after one of its four neighbors have downloaded all of the files to seed unchoke other neighbors. This is opposed to the original design of BitTorrent protocol. Originally featured on priority neighboring with the download/upload rate is the largest. However our algorithm is based on distance, and it ensures that the seeds sent parts to the nearest neighbor for the first time. If the upload rate of peers who have similar interests, similar to a closest peers with seed is probably the biggest thing to finish downloading first and becomes the seed. This makes the seeds spread across the center.

5. Evaluations

A. Experimental Setup

We used the Java-based PeerSim [32] as our simulation platform. PeerSim is a Peer-to-Peer simulator. It has been designed to be both dynamic and scalable. The engines consist of components which may be 'plugged in' and use a simple ASCII file based configuration mechanism which helps reduce the overhead. PeerSim can also work in two different modes [33] 1) cycle-based and 2) event-based. The cycle-based engine [34] is based on a very simple time scheduling algorithm and is very efficient and scalable. However, it has some limitations. PeerSim can achieve a network consisting of 10^6 nodes using the cycle-based engine. We create 1024 nodes in randomly with the simulate tools call's PeerSim simulates, and build them with in set (a) AS0, AS1, AS2, AS3and AS4. In set (b) AS0, AS1, AS2, AS3, AS4 AS5 and AS6. The Autonomous System topology is shown in Figure 3. 12 of these 1024 nodes are core nodes which only take charge of transmitting data. Bandwidth of the links between nodes that are equal between 50Mbps and 100Mbps fluctuant other 1,012 nodes as edge node. In the edge nodes 512 nodes, these are some of their peers had 512 peers all the seeds. Then we let a number of different kinds of peers, download files randomly every 10 seconds.



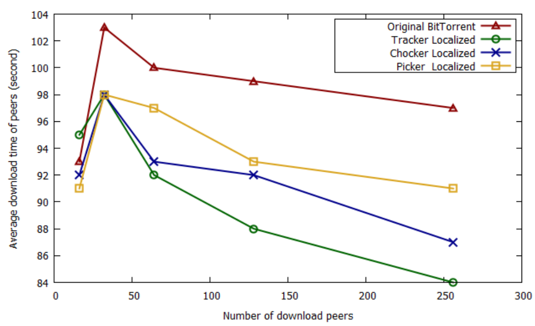
(a) The 5 of AS Topology. **(b) The 7 of AS Topology.**

Figure 3. The Autonomous System Topology in our simulations.

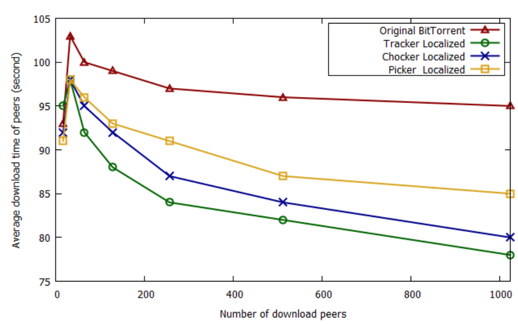
In our experiment, we set the 2 situation in all (a) 16 peers, 32 peers, 64 peers, 128 peers and 256 peers with (b) 16 peers, 32 peers, 64 peers, 128 peers 256 peers 512 peers and 1024 peers and take more than 5 times in every situation in each of the conditions, including the position of the nodes, the bandwidth of the link, the delays of the links and the download situation all the seeds and situation are generated randomly, except during certain special factors such as the number of nodes and the number of peers. Finally, we average the results obtained from the situation of a number of peers, and then compare them to those of the original algorithm.

B. Improving the Efficiency that User

We compare the performance that users in this situation these performance result include the average download time of peers for each localized algorithm and the time of all peers breaking downloading for each localized algorithm. The results are presented as follows.



(a) The average download time of peers for 5 of Autonomous System

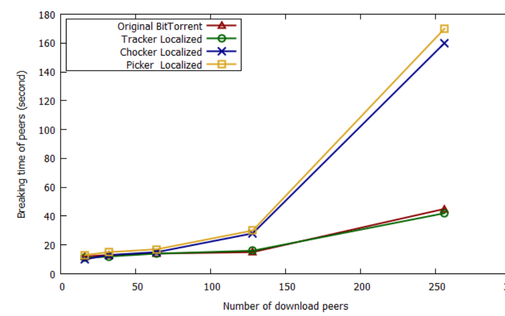


(b) The average download time of peers for 7 of Autonomous System

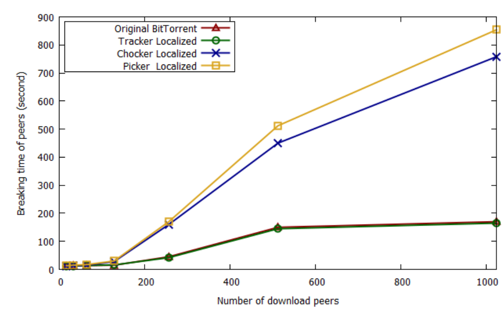
Figure 4. The average download time of peers for each localized algorithm.

Figure 4. shows the decrease download time of peers using chocker localized algorithm is more than using picker localized algorithm. When there are only a few download peers,the average download time decreases a little in chocker localized algorithms and piece picker localized algorithms. Because the peers often choose randomly among the neighbors closest to unchoke or download a piece of the closest which are effective in more download even when there are only a few seeds.

The download time average of peers for each localized algorithm. We can conclude that when we use the tracker localized algorithm, the downloading of peer decreases, but when there are few download peers, it doesn't decrease but increase a bit. The reason is that peers randomly selected neighbors to send information on the original BitTorrent. When there are a few download peers, the bandwidth of the links that are large enough to allow a faster download rate. However, after the trackers are trackers localized algorithm, the peers always select neighbors locally. Because the bandwidth of local links is smaller than that of inter-AS links, the data transmission rate is slower on local links. With the increase of download peers, the peers in the original BitTorrent still selected randomly which may lead to congestion of inter-AS links and performance degrading of the network. The tracker localized algorithm that makes localized traffic and use bandwidth more efficiently. Figure 5. Shows the breaking time of peers is minimum when the situation of picker localized algorithm, but the breaking time is maximum when the situation of tracker localized algorithm and increase the number of peer to download that can improve the efficiently for breaking time of all localized algorithm.



(a) The breaking time of all peers for 5 of Autonomous System

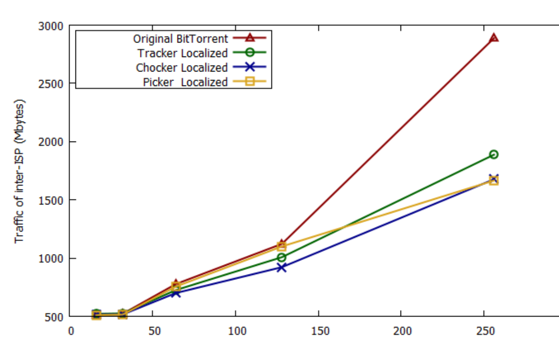


(b) The breaking time of all peers for 7 of Autonomous System

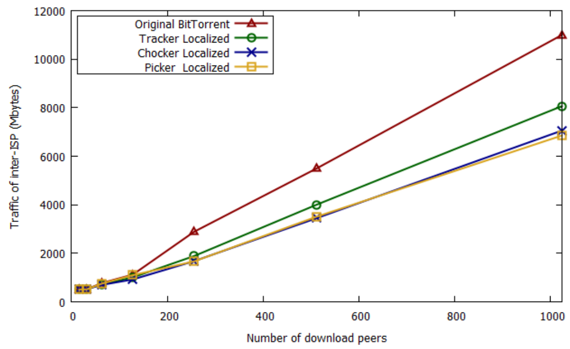
Figure 5. The breaking time of all peers downloading for each localized algorithm.

C. Improving the Efficiency that ISP

We compare the improving the efficiency that ISP for each localized algorithm in this situation these performance result include the traffic of inter-ISP for each localized algorithm. The results are presented as follows.



(a) The traffic of inter-ISP for 5 of Autonomous System



(b) The breaking time of all peers for 7 of Autonomous System

Figure 6. The traffic of inter-ISP for each localized algorithm.

Figure 6. shows the traffic of inter-AS links. The localized algorithms have optimized the traffic of inter-AS, localized algorithm to optimize traffic between AS and reduced to a similar extent. It is worth noticing that the extent of the decline in traffic is more obvious with the increase of the download peers. Because, the download for the duration of the amount of grain in all AS peers can download files time.

We also see that the algorithm selects only the reduction of traffic between inter-AS most obviously when there are many download peers. The reason is that the number of seeds in an AS is constantly increasing, which makes the picker localized algorithm optimize the performance more.

6. Conclusion

We studied in this paper the working of BitTorrent protocol and various mechanisms used to achieve optimal performance by the protocol. We presented the design, the simulation and the evaluation for improving the performance of New Technique to Improve BitTorrent Performance Based on Application Layer Traffic Optimization. BitTorrent is the most popular protocol that is widely used for P2P file sharing system. Nevertheless, BitTorrent protocol exposes some inefficient processes that are choosing neighbours and selecting pieces. These are mainly based on the knowledge of overlay topology without considering underlying internet topology that might cause of traffic problems such as the bottleneck at some point of network. Therefore, we propose the new approach for BitTorrent protocol with Tracker Localized Algorithm, Picker Localized Algorithm and Chocker Localized Algorithm which autonomous hops between peers are calculated to make efficient decisions.We conducted simulation of our scheme based on the PeerSim. The simulation results show that the peers provide better piece downloading on BitTorrent underlying topology and interacting among peers more efficiently. The Tracker localized algorithm outperforms the original BitTorrent for reducing the average downloading time of 20%. The breaking time of peers is the least when using Chocker localized algorithm. Additionally, the results show that ours scheme can decrease the peer-to-peer traffic and optimize traffic distribution for the whole network.

The experimental evaluation shows that our scheme node in the system, the situation that improvement of network infrastructure and can interact more efficiently. In addition, our scheme can help to decrease during inter-AS or inter-ISP traffic without sacrificing performance, optimize the distribution of traffic across the entire network and improve the quality of the user experience Peer-to-Peer users.

References

[1] B. Cohen. Incentives Build Robustness in BitTorrent. In Proc. of the 1st Workshop on Economics of Peer-to-Peer Systems, Berkeley, USA, June 2003.

[2] G. Urvoy-Keller and P. Michiardi, “Impact of inner parameters and overlay structure on the performance of BitTorrent,” INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Poceedings, pp. 1–6, April 2006.

[3] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. A. Hamra, and L. Garc’es-Erice, “Dissecting BitTorrent: Five months in a torrent’s lifetime,” in Proc. of Passive & Active Measurement Workshop, Apr.2004.

[4] X. Yang and G. Veciana, “Service capacity of peer to peer networks,” in Proc. of IEEE INFOCOM, Mar. 2004.

[5] D. Qiu and R. Srikant, “Modeling and performance analysis of BitTorrent-like peer-to-peer networks,” in Proc. of ACM SIGCOMM,Aug. 2004.

[6] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, “The BitTorrent P2P file-sharing system: Measurements and analysis,” in Proc. of International Workshop on Peer-to-Peer Systems, Feb. 2005.

[7] Bharambe, A. R., Herley, C., & Padmanabhan, V. N. (2005). Analyzing and improving bittorrent performance. Microsoft Research, Microsoft Corporation One Microsoft Way Redmond, WA, 98052, 2005-03.

[8] Guo, L., Chen, S., Xiao, Z., Tan, E., Ding, X., & Zhang, X. (2007). A performance study of BitTorrent-like peer-to-peer systems. Selected Areas in Communications, IEEE Journal on, 25(1), 155-169.

[9] D. Qiu and R. Srikant, “Modeling and performance analysis of BitTorrent-like peer-to-peer networks,” in Proc. 2004.

[10] Pouwelse J., Garbacki P., Epema D., and Sips H. “The BitTorrent P2P File-Sharing System: Measurements and Analysis.” In IPTPS’05. Feb. 2005.

[11] Raymond Lei Xia and Jogesh K. Muppala, Senior Member, IEEE, “A Survey of BitTorrent Performance",2010.

[12] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, “Analyzing and improving a BitTorrent networks performance mechanisms,” in Proc. IEEE INFOCOM, Apr. 2006, pp. 1–12.

[13] A. Legout, G. Urvoy-Keller, and P. Michiardi, “Rarest first and choke algorithms are enough,” in IMC ’06: Proc. 6th ACM SIGCOMM conference on Internet measurement. New York, NY, USA: ACM, 2006, pp. 203–216.

[14] J. Liu, H. Wang, and K. Xu, “Understanding peer distribution in global Internet,” IEEE Network Mag., 2010.

[15] Choffnes, D. R., & Bustamante, F. E. (2008). Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems. ACM SIGCOMM Computer Communication Review, 38(4), 363-374.

[16] L. Vu, I. Gupta, K. Nahrstedt, and J. Liang, "Understanding Overlay Characteristics of a Large-scale Peer-to-Peer IPTV System," ACM Trans. Multimedia, Computing, Commun. and Applicat., vol. 6, no. 4, pp. 1-24, Nov. 2010.

[17] J. Seedorf, S. Kiesel, M. Stiemerling, Traffic Localization for P2P-Applications: The ALTO Approach, In: Ninth International Conference on Peer-to-peer Computing (IEEE P2P 2009), pp. 171-177, Seattle, 2009.

[18] I. Rimac, V. Hilt, M. Tomsu, V. Gurbani, E. Marcocco, A Survey on Research on the Application-Layer Traffic Optimization (ALTO) Problem, RFC 6029 (Informational), 2010.

[19] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson, "One hop reputations for peer to peer file sharing workloads," in Proc. 2008.

[20] A. Legout, G. Urvoy-Keller, and P. Michiardi, "Rarest first and choke algorithms are enough," in Proc. ACM IMC, Oct. 2006, pp. 203–216.

[21] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in Proc. IEEE INFOCOM,Mar. 2005, vol. 4, pp.2235–2245.

[22] G. Dán and N. Carlsson, "Power-law revisited: A large scale measurement study of P2P content popularity," in Proc. IPTPS, Apr. 2010.

[23] G. Dán, N. Carlsson, "Centralized and distributed protocols for tracker-based dynamic swarm management. Networking," IEEE/ACM Transactions,2013, on, 21(1), 297-310.

[24] N. Carlsson, D. L. Eager, and A. Mahanti, "Using torrent inflation to efficiently serve the long tail in peer-assisted content delivery systems,"in Proc. IFIP/TC6 Netw., May 2010, pp. 1–14.

[25] L. Guanxiu, Y Suqi, and H. Xinli, "A Novel ALTO Scheme for BitTorrent-Like P2P File Sharing Systems." In Proceedings of the 2013 Third International Conference on Intelligent System Design and Engineering Applications, January 2013, pp. 135-139. IEEE Computer Society.

[26] G. Neglia, G. Reina, H. Zhang, D. Towsley, A. Venkataramani, and J. Danaher, "Availability in BitTorrent systems," in Proc. IEEE INFOCOM,May 2007, pp. 2216–2224.

[27] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurement, analysis, and modeling of BitTorrent-like systems," in Proc.ACM IMC, Oct. 2005, pp. 35–48.

[28] Y. Yang, A. L. H. Chow, and L. Golubchik, "Multi-torrent: A performance study," in Proc. IEEE MASCOTS, Sep. 2008, pp. 1–8.

[29] D. Menasche, A. Rocha, B. Li, D. Towsley, and A. Venkataramani,"Content availability and bundling in swarming systems," in Proc.ACM CoNEXT, Dec. 2009, pp. 121–132.

[30] R. S. Peterson and E. G. Sirer, "AntFarm: Efficient content distribution with managed swarms," in Proc. NSDI, May 2009, pp. 107–122.

[31] PeerSim page (3/9/2014) [Online]. Available: http://peersim.sourceforge.net/

[32] PeerSim Tutorial page (3/9/2014) [Online]. Available: http://peersim.sourceforge.net/tutorial1/

[33] PeerSim HOWTO page (3/9/2014) [Online]. Available: http://www.cs.unibo.it/babaoglu/courses/cas/resources/tutorials/peersim3.pd

[34] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and improving a BitTorrent networks performance mechanisms," in Proc.IEEE INFOCOM, Apr. 2006, pp. 1–12.

Nattee Pinthong

Received the B.S. and M.S. degrees in Information Technology from Mahanakorn University of Technology in 2006 and 2008, respectively. During 2009-2014, he studied in The Electrical Engineering Graduate Program, Faculty of Engineering, Mahanakorn University of Technology, Bangkok, Thailand. He is a member of the IEEE and ACM

WoraphonLilakiatsakun

Received the B.S. and M.S. degrees in Electronics Engineering from King Mongkut's Institute of Technology Ladkrabang in 1997 and 1996, and the Ph.D. degree in Telecommunication Engineering from the University of New South Wales, Australia, in 2004.

He is an Assistant Professor with the Faculty of Information Science and Technology, Mahanakorn University of Technology, Bangkok, Thailand.