On Improving Network Locality in BitTorrent-Like Systems

Yun-Chiu Ching[1], Ching-Hsien Hsu[1], and Kuan-Ching Li[2]

[1] Department of Computer Science and Information Engineering
Chung Hua University, Hsinchu, Taiwan 300, R.O.C.
{mali,robert}@grid.chu.edu.tw
[2] Department of Computer Science and Information Engineering
Providence University, Shalu, Taichung 43301, Taiwan
kuancli@pu.edu.tw

Abstract. The emerging Peer-to-Peer (P2P) model has become a very popular paradigm for developing Internet-scale systems. The BitTorrent is an example of P2P system for sharing resources, including files and documents. Owing to the peer does not have the capability of locality aware, it cannot differentiate its neighbors belong to which Internet Service Provider (ISP), file sharing results in a large number of cross-ISP traffic (about 70%). ISPs often control BitTorrent traffic by bandwidth limiting for reducing cross-ISP traffic. In this paper, we propose an adaptive peer collaboration strategy to reduce cross-ISP traffic without additional equipment and backup mechanism, which means decreasing the cost of additional equipment. Internal peers can collaborate indirectly. In peer collaboration strategy, a peer chooses most of its neighbors from internal ISP as itself, and only a few from external ISPs for reducing transfer of cross-ISP by biased neighbor selection. Second, in order to decrease redundancy, we employ Advanced Tracker (AT) to record the information of pieces that owned by each ISP. Finally, we adopt dynamic priority allocation for improving the file download time. Experimental results show that our peer collaboration strategy outperforms previous approaches, decreases redundancy and decreases the file download time remarkably.

Keywords: Peer-to-Peer, cross-ISP traffic, peer collaboration strategy, biased neighbor selection.

1 Introduction

BitTorrent [1, 11] file sharing system has become the most popular application in recent years. The early P2P file sharing systems are like Napster [2], Gnutella [3] and eMule [4]. Unlike the client-server model, BitTorrent divides a file into a number of equal-sized pieces, where each peer simultaneously downloads and uploads via its neighbors. BitTorrent has the special character, which the more users join BitTorrent, the faster download rate. Hence, file sharing creates a lot of BitTorrent traffic. The research [20] shows that BitTorrent generates cross-ISP traffic occupy most of Bit-Torrent traffic (about 70%). Each peer's neighbors are selected randomly from the

P. Mueller, J.-N. Cao, and C.-L. Wang (Eds.): Infoscale 2009, LNICST 18, pp. 58–75, 2009.

tracker, which indicates that each peer cannot decide its neighbors from the same ISP as itself. Additionally, each peer gets a lot of pieces from external peer [20]. Therefore, BitTorrent traffic has become a large number of cross-ISP traffic. ISPs often control BitTorrent traffic by bandwidth limiting for reducing cross-ISP traffic. However, bandwidth limiting will increase the file download time and worsen the user download experience, the fundamental concern of the ISP which is to improve the locality of BitTorrent traffic.

In this paper, we propose peer collaboration strategy, which includes three parts. First, we let a peer choose a lot of its neighbors from internal ISP as itself, and only a few from external ISPs by modifying trackers and clients [9]. This conception is doing a similar grouping BitTorrent peers into clusters [10] and let most of pieces of exchange rely on internal ISP, and only a few of pieces rely on external ISP, improving traffic locality in BitTorrent and reducing the times of cross-ISP. Besides, the approach of modifying trackers and clients can come to biased neighbor selection [9] without additional equipment, it conform our spirit that as economic as possible and avoid the equipment had a breakdown, and decreasing cost of equipment. Moreover, the biased neighbor selection is key to the success of reduces cross-ISP traffic and rely on BitTorrent adopts the local rarest first (LRF) algorithm [11, 19] as the piece selection strategy, where each peer downloads piece which is least replicated pieces among its neighbors.

The purpose of BitTorrent peers into clusters is to solve redundancy problem [9] and it is valid for reducing cross-ISP traffic. Owing to each peer's neighbors are selected randomly from the tracker, the redundancy is very high as start to share the file. The optimal redundancy is 1 [9], which means each piece will only be transferred once from initial seed to other ISPs. Even if there is no backup mechanism, our goal in this paper is to let redundancy down as low as possible (close to 1). We present peer collaboration strategy, the lower redundancy, inspired by FPFR [18] and MOB [10]. They are both multicast approach and their redundancy are 1. The distribution of MOB in grid is similar to BitTorrent. Unlike the BitTorrent, MOB adds teamwork among the nodes of a cluster to improve collaboration. BitTorrent employs incentive mechanisms such Tit-For-Tat (TFT) [11, 19] to improve ratio of upload. However, BitTorrent is based on the personal benefit. Many users always leave as finishing their download and some of users are free-riding [13]. In order to decrease redundancy, each peer needs to collaborate. For example, each peer downloads different piece from external ISPs and does not download the piece if it is in internal ISP already. BitTorrent has no teamwork among peers according to the above-mentioned statement. We propose a strategy and it let each peer know whether the piece in internal ISP to avoid downloading the redundant piece, which is exist in internal ISP. Thus, second part of peer collaboration strategy is that decreasing redundancy is close to 1. We employ Advanced Tracker (AT) to maintain a list, which records the information of pieces that owned by each ISP presently. Each peer can know whether the piece in internal ISP in terms of a list, and needs to get the piece by external peer if there is no that piece in internal ISP.

Peer collaboration strategy may cause some peers are waiting for the internal piece rather than external piece, and increase the file download time. On the other hand, all the download process of BitTorrent, especially in the first piece scenario and last piece scenario, because of the choke algorithm the paradox of supply and demand exists in

BitTorrent [16], which is increasing the file download time. Finally, we adopt dynamic priority allocation to decrease the file download time.

The rest of this paper is organized as follows. We present an overview of BitTorrent and the relate work of cross-ISP traffic in section 2. We present the preliminaries in section 3. We present peer collaboration strategy in section 4 and give the experimental evaluation in section 5. We conclude this paper and future work in section 6.

2 Related Work

2.1 BitTorrent Overview

The BitTorrent P2P system has become the most popular sharing resources, and its kernel source code had been written by Bram Cohen in 2002[11]. Unlike the HTTP/FTP model, a large number of files were shared by BitTorrent via Internet without increasing the loading of publisher's server and bandwidth. The performance of BitTorrent outperforms the traditional approaches remarkably. The main idea of BitTorrent is that each peer simultaneously downloads and uploads. Therefore, the download rate does not restrain by publisher's upload bandwidth, and much more users join the torrent can supply more upload bandwidth. BitTorrent is that a shared file is divided into a number of equal-sized pieces (typically 256 KB in size), and each piece is split in sub-pieces (typically 16 KB in size) to avoid a delay among pieces being sent, and always keeping some number (typically 5) requests pipelined at once. Sub-pieces are the transmission unit on the network, but the protocol only accounts for transferred pieces. Each peer simultaneously downloads and uploads after get first piece from its neighbors, and each peer is a serve as well as a client. Therefore, BitTorrent can distribute the pieces quickly and it has a higher transfer rate than the traditional approaches, and it also does not increase the loading of publisher's server.

Figure 1 shows that the file download process of BitTorrent is as follows. (1) A user downloads a metadata file (called the torrent file) was generated by publisher from a web server, which contains IP address of tracker and the SHA-1 hash values of each piece, and the piece size and so on. (2) A user starts the BitTorrent client software to join and contacts the tracker as a new peer. Tracker is s central server, which keeps track of all peers downloading the file. The new peer requests a part of peers from the tracker, which responds to the new peer with a list of randomly chosen peers (typically 50), then the new peer attempts to establish connection with these peers as its neighbors. (3) The new peer starts to exchange pieces with its neighbors. BitTorrent let tracker keep an up-to-date state, and every 30 min each peer reports to the tracker its state. Figure 1 shows that the Seed is initial seed, which means the file publisher, and there are two types of Peer, namely leechers and seeders. Leechers also called downloaders, are peers who only have a part (or none) of the pieces of the file, which seeders are peers who have all pieces of the file. Leechers simultaneously download and upload pieces, and seeders only upload piece. A leecher turns into a seeder when it obtains all the pieces. At this moment, a seeder can leave BitTorrent or stay online to upload pieces continually. When the number of its neighbors dips below the threshold (typically 20), the peer again contacts the tracker to obtain a list of additional neighbors.

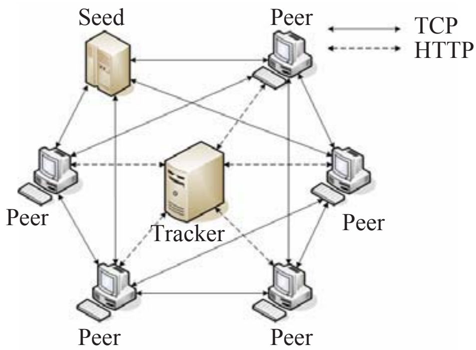


Fig. 1. The framework of BitTorrent

BitTorrent does not the central sharing resources, so it needs a great algorithm to improve fairness and optimizing the whole download rate. BitTorrent adopts choke algorithm includes TFT and Optimistic Unchoke (OU) as the peer selection strategy, where each peer can upload pieces to five peers, so each peer has five upload quotas, among which each peer allocates four quotas for the TFT and the fifth for the OU. Process of Choke algorithm as follows, every 10 seconds, a peer employs the TFT to evaluates which of its interested neighbors have been giving pieces to it with the high rates and the four highest peers are preferred to unchoke. Every 30 seconds, a peer employs OU to randomly unchoke from the remaining set of peers. OU allows boot-strap new peers and explores other peers which potentially upload to it with the higher rates. Existing BitTorrent employs the static quota allocation to fixedly allocate upload quotas for the TFT and OU. The peer starts to send <request> message to request pieces when the other side sends <unchoke> message to it. How to choose the piece is also very important. A poor piece selection may cause the last pieces problem [11,19] and let peers obtain the incomplete file, and furthermore it may cause the low download rate or increase the loading of upload peer. BitTorrent employs four strategies as piece selection as follows.

(1) **Strict Priority:** In BitTorrent, a complete piece is the exchange unit for file sharing, but sub-pieces are the transmission unit on the network. Hence, once a single sub-piece has been requested, the remaining sub-pieces from that particular piece are requested before sub-pieces from any other piece.
(2) **Rarest First:** Under this strategy, a peer downloads the piece, which is least replicated among its neighbors is chosen first. Even if there are no seeds in BitTorrent, the peer may get piece from its neighbors to avoid the last pieces problem.
(3) **Random First Piece:** If a new peer has nothing to upload and exchange, it is important to get one piece as quickly as possible, and the new peer starts to employ the rarest first policy until it gets first piece.
(4) **Endgame Mode:** When a user has most of pieces already, which means only a few of pieces does not obtain, the peer sends requests for all sub-pieces to its neighbors. Cancels are sent for sub-pieces as one of its neighbors starts to upload the sub-pieces, to avoid redundant sends and the end of the file can download quickly.

As shown in Figure 1, there are two types of protocol in BitTorrent, one is the connection between peers and tracker, its protocol is based on HTTP, and the other is the connection among peers, its protocol is based on TCP.

2.2 Existing Studies on Cross-ISP Traffic

There are many previous approaches to solve issue about cross-ISP traffic, some of them need higher cost of equipment, and P2P systems and ISPs need collaboration closely [5,6,7,22], others only spend low cost and P2P systems collaborate with ISPs indirectly [9,12,20]. The purposes of these approaches are improving the file download time and decreasing redundancy. In this paper, a peer inside the same ISP we called internal peer, conversely, we called external peer.

In [5], positional at the gateway of ISP to the Internet, a cache stores pieces sent by external peers to internal peers, and as internal peer wants to get the piece from external peer, the cache interrupts and sends the copy to internal peer. In [6,7], ISP supplies an oracle service to users. ISP employs the equipment to manage each peer to avoid a lot of BitTorrent traffic. Likewise, users can obtain a great download rate. However, they need the user supplies the oracle with a list of possible neighbors, and then oracle ranks them according to certain criteria such as distance and bandwidth. Another approach is to employ the gateway peer [20], and it is the only peer inside internal ISP that can connect to external peers. But gateway peer needs high upload bandwidth to avoid increasing download time [9], and gateway peer needs to keep stable performance. P2P traffic shaping devices [9] can intercept and modify the responses from the tracker to the peer, and let peers drive to biased neighbor selection. P4P [22] supplies the interactive interface to P2P system and ISP, and it allows ISP to manage the underlying physical network and supplies the real- time network to P2P system. Furthermore, it can integrate network command into P2P system, improving the network utility ratio and performance of P2P system. The content distribution networks (CDN) [12] to drive biased neighbor selection without collaboration between users and ISPs. A peer only needs to have the ability to perform local DNS queries for CDN names, and then it can find its neighbors. Modifying trackers and clients is another approach [9], and this approach allows each peer to get most of peers from internal ISP, only a few from external ISP, and it is a direct and efficient approach. Using bandwidth limiting to control BitTorrent traffic is the simple and direct approach, but it is not efficiency.

These approaches are the same principle, which let BitTorrent peers into clusters and improving the traffic in internal ISP. Consequently, several users can decrease the file download time and ISP does not employ bandwidth limiting to reduce cross-ISP traffic. However, various approaches need to raise the cost of hardware except modifying trackers and clients and CDN [12]. For example, Cache, P2P traffic shaping devices, Oracle service and P4P portal, and it needs the particular peer to have highest bandwidth and stable performance, such as gateway peer. In this paper, our peer collaboration strategy can reduce cross-ISP traffic without sacrificing system performance and is based on the premise that as economic as possible. Besides, our peer collaboration strategy only needs to modify original framework of BitTorrent, and it does not need additional equipment. Experimental results show that our peer collaboration strategy can decrease redundancy and decrease the file download time efficiently.

3 Preliminaries

Generally, BitTorrent goes through three stages in its life: flash crowd, steady state and winding down [8, 15, 17, 21]. Among the three stages, flash crowd stage creates a lot of traffic and is the most challenging for ISPs to control, so we focus on the flash crowd stage in this paper. We assumed that each peer leaves as finishing its download, only initial seed always stays online until terminating the simulation, and initial seed does not employ the biased neighbor selection. In this paper, the main evaluation criteria are redundancy and download time. Improving traffic locality can reduce cross-ISP traffic, but the lower redundancy, the lower cross-ISP traffic.

In section 2, we can find three problems about original BitTorrent. (1) Each peer's neighbors are selected randomly from the tracker, and file sharing creates a lot of BitTorrent traffic. (2) BitTorrent adopts the LRF algorithm as the piece selection, and it cannot avoid redundancy is increasing even if we solve problem (1). (3) The whole download process of BitTorrent in the first piece scenario and last piece scenario, because of the choke algorithm the paradox of supply and demand exists in BitTorrent [16], which is increasing the file download time. We present the peer collaboration strategy as follows, and it has three parts, which solve the above problems.

(1) **Biased neighbor selection:** According to the above-mentioned statement, each peer's neighbors are selected randomly from the tracker. Hence, file sharing may create a lot of cross-ISP traffic. ISPs employ bandwidth limiting in order to reduce cross-ISP traffic to decrease cost. Therefore, we may not have a great performance even if we use the high bandwidth. Some factors affect the performance such as network congestion, ACK delay, and smaller bandwidth, etc. Likewise, ISPs also need to control BitTorrent traffic. In order to create a win-win situation, using biased neighbor selection is a direct and efficient approach, and ISPs do not worry about the cross-ISP traffic all the time but users have a better download experience. We adopt the approach of modifying trackers and clients to finish biased neighbor selection [9]. Modifying tracker we called Advanced Tracker (AT). When a new peer join the torrent. According to the new peer's ISP, AT let the new peer chooses M-N neighbors from internal ISP as itself, and only N neighbors from external ISPs, where M is the maximum connection of peer, and N is the number of external peers.

(2) **Unique piece selection:** BitTorrent adopts the LRF algorithm as the piece selection. Even if BitTorrent had been employed the biased neighbor selection, each peer may get the redundant piece, which means the piece had downloaded by some internal peer. Because BitTorrent only employs the LRF algorithm as the piece selection and is affected by choke algorithm. Our method as follows, we let AT maintain a table, we called Global Unique Piece Table (GUPT). It records the information of pieces that owned by each ISP and peers can observe the GUPT to avoid downloading the redundant piece. Therefore, each peer get the piece from external peer is unique inside internal ISP every time. Internal peers can collaborate indirectly, and decrease redundancy significantly. Also, we present the two approaches to solve the problem that the paradox of supply and demand in terms of GUPT. We detailed introduce it in section 4.2.

(3) **Dynamic priority allocation:** In unique piece selection, GUPT will probably make some peers are waiting for the inside piece rather than outside piece, so increasing download time. On the other hand, the whole download process of BitTorrent in the first piece scenario and last piece scenario, because of the choke algorithm the paradox of supply and demand exists in BitTorrent [16], which is increasing the file download time. The advantages of Dynamic Quota Allocation (DQA) [16] are preserved in our dynamic priority allocation, and we let OU allocate quota to external peer, improving the download rate via optimizing the upload quota utility ratio. It is noted that we do not make the backup mechanism, dynamic priority allocation needs each peer to upload more 2-4 pieces in terms of it follows the DQA, and it can complementary to the problem that ISP lost pieces. Dynamic priority allocation can let pieces spread to different ISP as quickly as possible because of external peer has a higher priority.

4 Peer Collaboration Strategy

In this section, we present our Peer Collaboration Strategy (PCS). We explain how internal peers to collaborate indirectly in terms of the whole framework and flowchart in section 4.1 and section 4.2. We propose an approach to improve the download time in section 4.3.

4.1 Biased Neighbor Selection

The basic approach to solve the problem of cross-ISP traffic is grouping BitTorrent peers into clusters so as to reduce the connection of external ISP. In BitTorrent, each peer's neighbors are selected randomly from the tracker, and most of neighbors spread to different ISPs. Hence, file sharing creates a lot of cross-ISP traffic. Internal peers can let a lot of exchange inside internal ISP, and only unique piece rely on external peer after BitTorrent peers into clusters.

In order to let BitTorrent peers into clusters, we choose the approach of modifying trackers and clients to drive biased neighbor selection [9], and this is a direct and efficient approach. Modified tracker we called Advanced Tracker (AT). When a new peer join the torrent, According to the new peer's ISP, AT let the new peer choose M-N neighbors from internal ISP as itself, and only N neighbors from external ISPs, where M is the maximum connection of peer, and N is the number of external peers, and we set N = 1.

Figure 2 shows that the biased neighbor selection is different from random neighbor selection. As shown in Figure 2, biased neighbor selection reduces a lot of external connection significantly. Thus, it does not need additional equipment to drive biased neighbor selection and conform to the principle of economic. Also, the redundancy will decrease.

However, BitTorrent adopts the LRF algorithm as the piece selection. The rarest piece cannot present the unique piece inside internal ISP. Therefore, each peer has many chances to get redundant piece in external ISP even if BitTorrent peers into clusters already. The choke algorithm also acts the important roles, and they are keys to reduce the redundancy. In section 4.2, we propose a strategy to restrain the redundancy

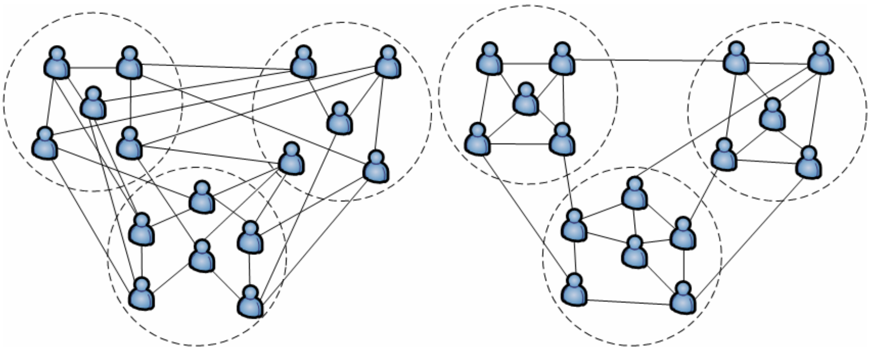


Fig. 2. Random neighbor selection vs. biased neighbor selection

efficiently. Let each peer get the piece from external peer is unique inside internal ISP every time, besides an exception. For example, several peers may simultaneously leave, so there is no seed in internal ISP, and ISP lost pieces. At that time, the peer needs to send request to external peer if initial seed does not in internal ISP.

4.2 Unique Piece Selection

4.2.1 Global Unique Piece Table

In section 4.1, we use AT to drive biased neighbor selection. However, we find some scenario, which would increase redundancy after biased neighbor selection.

(1) **LRF algorithm:** The definition of rarest piece is least replicated among its neighbors [19], which indicates rarest piece is not unique in internal ISP except the rarest piece first travels into internal ISP. Once the rarest piece travels into internal ISP which it had downloaded by internal peer, the redundancy will increase gradually. The LRF algorithm does not use to solve the problem of choosing unique piece. Thus, we need a strategy to get unique piece from outside.

(2) **Choke algorithm:** If some peer is interested in internal peer and external peer simultaneously, the redundancy may increase via choke algorithm. For example, internal peer sends <choke> message and external peer sends <unchoke> message in terms of TFT. The redundancy will increase as the particular piece had been downloaded before.

(3) **ISP lost pieces:** Internal peer needs external peer to get some rarest piece again when ISP lost pieces happened, which was due to several peers may simultaneously leave and peers include seed or initial seed is not here. ISP lost piece cannot prevent, besides the system uses the backup mechanism [5] or network coding [14]. The redundancy will increase deservedly when ISP lost pieces.

(4) **Synchronal request:** When there are no records on GUPT (in internal ISP, a piece had not been downloaded yet), internal peers may download pieces via external peers simultaneously because no records on GUPT for references.

We present the second part of peer collaboration strategy is that we let redundancy close to 1. Internal peers cannot collaborate to decrease redundancy according to (1)(2)(3)(4). Our unique piece selection can solve (1)(2). We organize (3)(4) into exception because it cannot avoid the redundancy is increased in our strategy. We do not discuss the matter and allow the redundancy is increased in (3)(4). In unique piece selection, our method is that we let AT maintain GUPT to record the information of pieces that owned by each ISP. Initially, the file was allocated by initial seed and spread it around. When a peer has received <bitfield> or <have> message from internal peer, it does not check the GUPT and it only uses the incentive mechanisms of original BitTorrent. When a peer has received the <bitfield> or <have> message from external peer, it checks the record of the particular piece whether it had been recorded on GUPT. If there is no record about the particular piece on GUPT, the peer sends <interested> message and waiting to have the <unchoke> message with the opposite side. Conversely, sends <not interested> message directly to inform the opposite side. If the peer is unchoked by the opposite side, it informs AT to record the particular piece on GUPT as it has done to request the piece. Figure 3 depicts the flowchart of unique piece selection, and this scheme can prevent the redundant pieces from external ISPs except ISP lost piece and synchronal request.

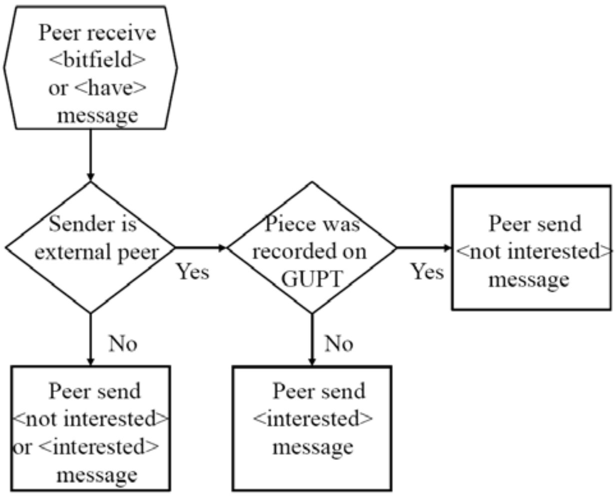


Fig. 3. Flowchart of unique piece selection

4.2.2 Solution of ISP Lost Pieces Problem

Some internal peers only rely on external peer to get the piece from external peer when ISP lost pieces happened. Some peers cannot get piece from external peer because of the record on the GUPT. Peers cannot complete their download except initial seed inside the same ISP. Therefore, ISP lost pieces happened, our solution as follows.

(1) **Peer contacts AT again:** Generally, ISP lost pieces is due to several peers may simultaneously leave, and let ISP lost some rarest piece. In BitTorrent, when the number of its neighbors dips below the threshold (typically 20), the peer again contacts the tracker to obtain a list of additional neighbors. If the peer get piece at once after reconnection, which indicates that locality lost piece rather than whole ISP. And if no action in the long time, which indicates that the peer only rely on external peer to get the piece (maybe external peer is a seed or lecher who has lost

piece, even initial seed). Our method as follows, the peer does not consult GUPT anymore as the peer does not send any <interested> message after get a list from AT again in 30 seconds.

(2) **GUPT records the number of piece:** GUPT also record the number of each piece. Our method as follows, each peer informs AT to count the number of piece as get the piece from internal ISP, and subtract the one unit with all pieces as leaves, if the number of some piece is zero, which indicates a peer can request the loss piece from external peer again.

4.3 Dynamic Priority Allocation

It is not difficult to find issue in unique piece selection scenario, which GUPT will probably make some peers are waiting for the inside piece rather than outside piece, so increase download time or ISP lost pieces happened. On the other hand, the previous studies show that the whole download process of BitTorrent in the first piece scenario and last piece scenario, because of the choke algorithm the paradox of supply and demand exists in BitTorrent [16], which is increasing the file download time. Each new peer can only rely on the upload peer uses the OU algorithm to randomly upload to one of these new peers, which causes many new peers to be starving. Unfortunately, BitTorrent adopts the static quota allocation, TFT cannot allocate pending upload quota to OU. In DQA, OU can use more upload quotas that upload quotas are located at pendency, to optimize the upload quota utility ratio and solve the problem that the paradox of supply and demand.

Although there are two approaches to solve ISP lost pieces in unique piece selection. We prefer to adopt the zealous approach to prevent ISP lost pieces. First, Dynamic Priority Allocation (DPA) adopts the strategy of DQA let the system allocate the piece to neighbors more quickly. DQA let each peer has a smooth, average and steady download rate. The new peer follows procession to download, it can avoid most of peers download very slowly, and only a few of peers download very quickly. It is easy to cause ISP lost pieces via several peers simultaneously leave. Each peer has a smooth download rate, which indicates it has smaller standard deviation that the interval of the download time among peers. Thus, a large amount of seeds can be produce in a short time. Moreover, we preserve advantages of DQA that adopt a weight allocation scheme to adaptively allocate upload quotas for TFT and OU in our dynamic priority allocation and let external peer has a higher priority to allocate the quota of OU in terms of the different piece is allocated to different ISP as quickly as possible and is suited to the our condition (cross-ISP). Formulas (1) to (8) are the deduction of DQA. Q denotes the total upload quotas, Q_{tft} and Q_{ou} denotes the upload quota for the TFT and OU separately. Furthermore, we set Q_{ou} divided into Q_{urg} and Q_{norm}. They are allocated by the following formula:

$$F = \left|P_{fast}\right| \cdot \alpha \tag{1}$$

$$L = \left|P_{low}\right| \cdot (1-\alpha) \tag{2}$$

$$Q_{tft} = \frac{F}{F+L} \cdot Q \tag{3}$$

$$Q_{ou} = \frac{L}{F+L} \cdot Q \tag{4}$$

$$U = |P_{urg}| \cdot \beta \tag{5}$$

$$N = |P_{norm}| \cdot (1-\beta) \tag{6}$$

$$Q_{urg} = \frac{U}{U+N} \cdot Q_{ou} \tag{7}$$

$$Q_{norm} = \frac{N}{U+N} \cdot Q_{ou} \tag{8}$$

Where $|P_{fast}|$ denotes the number of the set P_{fast} and $|P_{low}|$ denotes the number of the set P_{low}. Likewise, $|P_{urg}|$ denotes the number of the set P_{urg} and $|P_{norm}|$ denotes the number of the set P_{norm}. α denotes the weight of P_{fast} and β denotes the weight of P_{urg}, and we set $\alpha = 0.7$ and set $\beta = 0.8$ in terms of dynamic quotas allocation, which means P_{fast} has a higher priority than P_{low} and means P_{urg} has a higher priority than P_{norm} separately.

We use formula (7) to allocate the urgent upload quotas Q_{urg} divided into Q_{ex} and Q_{in}. In formula (9)~(12), $|P_{ex}|$ denotes the number of the set P_{ex} and $|P_{in}|$ denotes the number of the set P_{in}. γ denotes the weight of P_{ex}, and we set $\gamma = 0.7$, which means P_{ex} has a higher priority than P_{in}. Dynamic priority allocation mainly solved the download time and ISP lost pieces problem.

$$E = |P_{ex}| \cdot \gamma \tag{9}$$

$$I = |P_{in}| \cdot (1-\gamma) \tag{10}$$

$$Q_{ex} = \frac{E}{E+I} \cdot Q_{urg} \tag{11}$$

$$Q_{in} = \frac{I}{E+I} \cdot Q_{urg} \tag{12}$$

5 Experimental Evaluation

This section presents our experimental evaluation. First, we introduce our simulation environments and methodology, and then analyze our experimental results to prove PCS is effectiveness.

5.1 Methodology

ISP traffic redundancy and download time are the main evaluation criteria in our simulation experiments. The term ISP traffic redundancy means the average number of times each piece crosses the ISP, until all peers inside the ISP finish their download. The lowest redundancy is 1. The highest redundancy is R, where R is the number of peers inside the ISP. Measurement of the file download time uses the cumulative distribution function (CDF). We design a discrete event driven simulator for BitTorrent, which primarily simulates the peer behavior such as (1) peer joining/leaving, (2) peer obtains a list and connects its neighbors, (3) piece transfer, (4) peer reports the tracker periodically, and the main incentive mechanisms such (5)choke algorithm, (6)LRF algorithm in the flash crowd stage. The framework of network consists of 14 ISPs, they are assumed to be completely connected and we combine bandwidth limiting with our simulation environments. It is noted that we ignore the simulation of the underlying physical network characteristics such as propagation delay, congestion control and flow control, etc. This approach also assumes idealized performance of TCP, and does not model the dynamics and traits of TCP implementations. Each ISP have 50 peers, and we simulate 700 peers including 1 initial seed and use one tracker, and the initial seed bandwidth is 400Kbps and does not use biased neighbor selection.

Our simulation environments examine two network settings, the upload/download bandwidth for peers (100Kbps/1Mbps) in the homogeneous network and the heterogeneous contain a high-bandwidth peers (we called extra peers) with the homogeneous, the upload/download bandwidth for extra peers (1Mbps/1Mbps) in the heterogeneous. We assume all extra peers have point-to-point links with each ISP and also with each other. Each peer leaves as finishing its download, only initial seed always stays online until terminating the simulation. The shared file with size of 64 MB is divided into 2000 equal-size pieces with each piece size 32KB. Each peer has five upload quotas, four quotas for the TFT algorithm and the fifth quota for the OU algorithm. In DPA, we set $\alpha = 0.7$, $\beta = 0.8$ and $\gamma = 0.7$. Biased neighbor selection (BNS), GUPT and DPA are to combine to the PCS. Our simulation environments and previous approach [9] are the same because we primarily compare with it.

5.2 Performance Analysis

PCS compare with previous approaches in terms of redundancy and the file download time. In [9], we know only bandwidth limiting cannot restrain the redundancy and decrease the file download time. Table 1 shows that we set ISP bandwidth from 2.5Mbps to 500Kbps, the redundancy still higher (about 21), and the redundancy is decreased only slightly between 1.5Mbps and 500Kbps. It appears that ISP bandwidth limiting cannot reduce cross-ISP redundancy anymore, and the download time increase 2.6 factor. The performance can does better as long as we use the BNS.

Table 1. Normalized download time and ISP traffic redundancy under ISP bandwidth limiting in the homogeneous networks

ISP bandwidth limiting	Time	ISP traffic redundancy
No limiting	7932s	47.13
2.5Mbps	8675s	31.47
1.5Mbps	10891s	25.23
500Kbps	18738s	21.95

Figure 4 depicts the ISP traffic redundancy of each approach without using ISP bandwidth limiting in the homogeneous networks. As shown in Figure 4, the redundancy of BitTorrent is about 47 (total internal peer is 50, so the highest redundancy is

50). It is noted that each peer almost downloads piece from external peer, it is a poor traffic locality inside the ISP. The redundancy immediately down to 4 after the original BitTorrent uses the BNS. Our PCS is based on GUPT and it combines with GUPT, which indicates that peers can collaborate indirectly, so the redundancy of PCS can keep to about 3. Factors of increasing redundancy are only ISP lost pieces and synchronal request. Our PCS has unique piece selection strategy, so the redundancy is lower than BitTorrent only uses the BNS in the homogeneous. Moreover, we combine BNS and GUPT with DQA and DPA separately to examine. These strategy are used to decrease download time, there is no effect upon the redundancy in this state except do not use BNS. Also, Figure 4 shows that the redundancy is almost the same as the strategy adds DQA and DPA. The average redundancy of DPA is lightly better than DQA, the key is DPA can let pieces spread to different ISP as quickly as possible because of external peer has a higher priority, which indicates that each peer has a smooth, average and steady download rate to avoid a few of peers leave as finishing their download and ISP lost pieces happened.

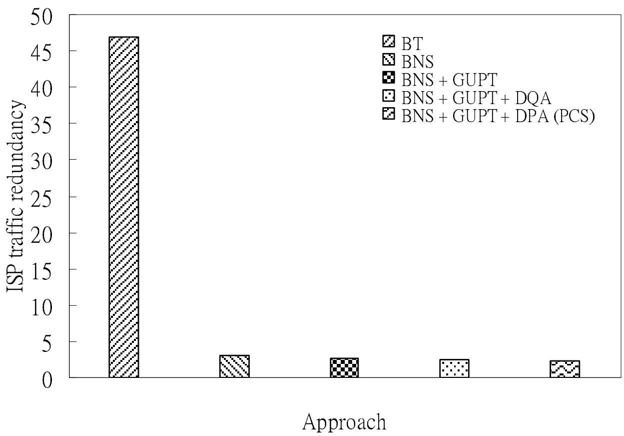


Fig. 4. ISP traffic redundancy of each approach in the homogeneous networks

Figure 5 depicts the file download time of each approach in the homogeneous. We can find our PCS outperforms BNS. For example, in 4000s, the PCS has about 95% peers completing all pieces while the BNS has only 5% peers. DPA is based on DQA, so each peer can maintain smooth download time among all pieces includes in the first piece scenario and last piece scenario. As shown in Figure 5, also we find the DPA and DQA allocate pieces quickly in the early phase, and each peer keeps smooth download rate. Several peers finish their download simultaneously in a short time. They have a great performance about the file download time, and they are faster than BNS and original BitTorrent. Besides, DPA can let pieces spread to different ISP as quickly as possible because of external peer has a higher priority, DPA is faster than DQA by about 1.1 factor and faster than BNS by about 1.6 factor, and faster than original BitTorrent by about 1.77 factor.

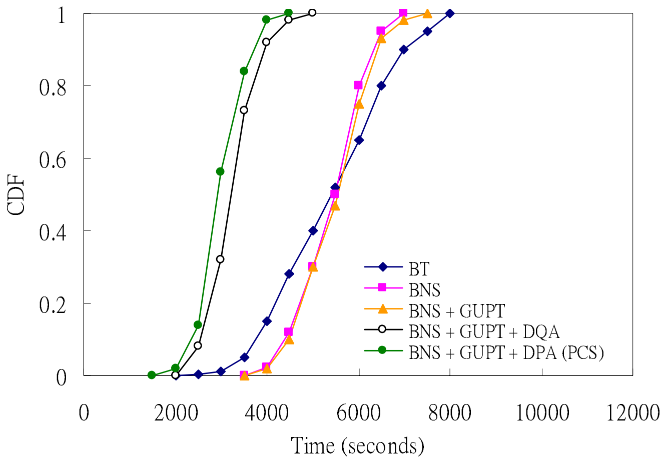


Fig. 5. Download time of file in the homogeneous networks

Figure 6 depicts the effect upon the redundancy of additional extra peers in the heterogeneous, we add extra peers from 10 to 50 and extra peers do not use BNS, and leave as finishing their download. As shown in Figure 6, we find the redundancy will increase gradually with BNS because of additional extra peers. PCS combines with GUPT, so it can restrain the redundancy. The redundancy of PCS keeps 5 as adds extra peers until 50, but the redundancy of BNS is >10 already.

Figure 7 shows that we contrast the download time of all approaches in heterogeneous networks, and the performance of all approaches in the heterogeneous networks outperform in the homogeneous networks due to extra peers have the high upload

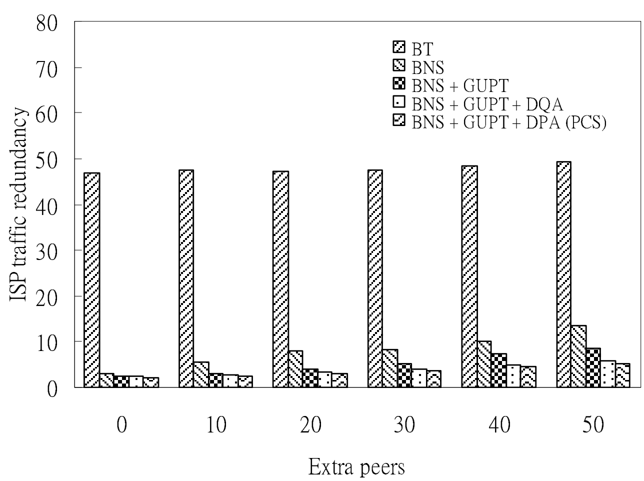


Fig. 6. ISP traffic redundancy of approach in heterogeneous networks

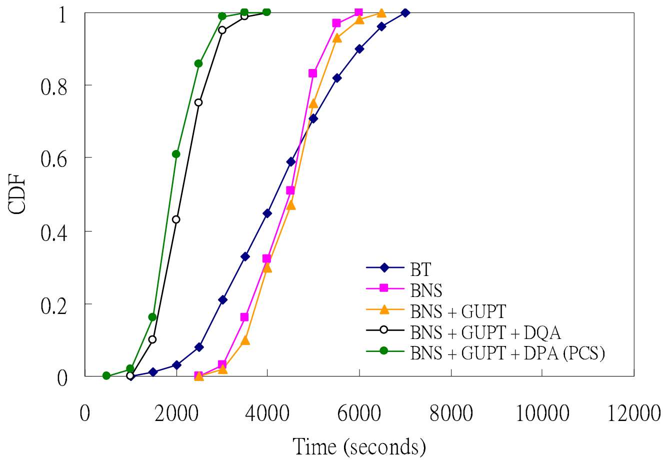
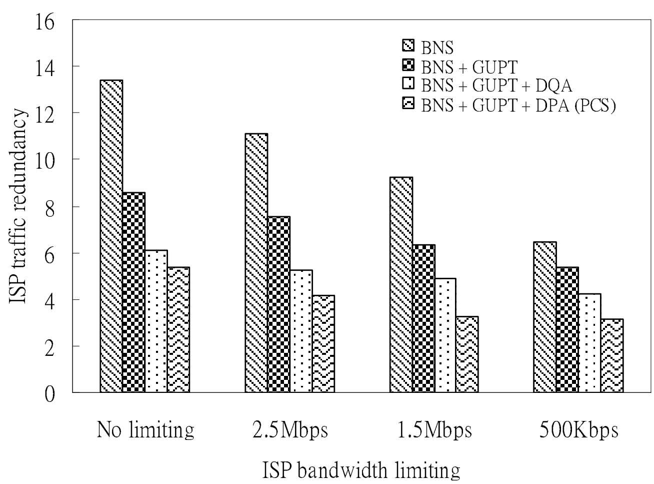


Fig. 7. Download time of file with 50 extra peers in heterogeneous networks

bandwidth, and PCS has the fastest download time. We can find our PCS not only restrain the redundancy as shown in Figure 6, but the download time of PCS still outperforms others in Figure 7.

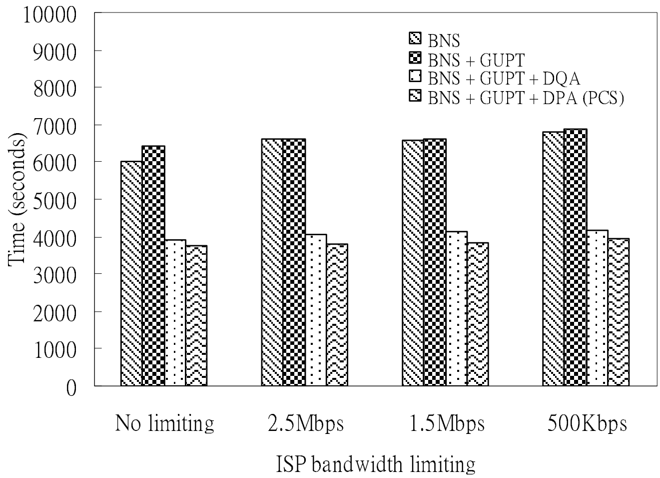
Figure 8(a) shows that we let all approaches add bandwidth limiting to reduce redundancy except original BitTorrent. We look into the performance of BitTorrent in the heterogeneous networks through bandwidth limiting. Because of the reality of truth, ISP needs to restrain the P2P traffic by bandwidth limiting. We set bandwidth limiting from 2.5Mbps to 500Kbps.



(a) ISP traffic redundancy

Fig. 8. Combination of bandwidth limiting with 50 extra peers in the heterogeneous networks

As shown in Figure 8(a), all approaches can reduce redundancy through bandwidth limiting, and PCS has the lowest redundancy. Figure 8(b) shows that PCS maintains the fast download rate after PCS adds bandwidth limiting. The file download time of PCS is increased only slightly (about 5%), and it has a fewer effect upon the users. But BNS increase by about 13%.



(b) Download time

Fig. 8. (*Continued*)

6 Conclusions and Future Work

We have presented Peer Collaboration Strategy (PCS), which integrates the advantage of biased neighbor selection (decrease redundancy) and DQA (decrease the file download time) into our strategy, and combines with GUPT to let redundancy down. PCS can ensure redundancy is controlled, besides the status of ISP lost pieces. Experimental results show that our peer collaboration strategy can reduce cross-ISP traffic without sacrificing system performance, and we have to modify original framework of BitTorrent, and it does not need additional equipment and backup mechanism. In terms of the user perspective, they only care about the file download time and if the seed does not exist. Therefore, a lot of previous approaches only focus on the file download time and most of the ideas are based on personal demand, and they do not consider cross-ISP traffic. However, the reality of truth, a large number of cross-ISP traffic will let ISPs adopt bandwidth limiting, and it brings the low download rate and redundancy out of control. In this paper, we consider the whole performance of BitTorrent in terms of the ISPs perspective, and we focus on the reduce cross-ISP traffic and the next is the whole performance of BitTorrent. Improving the BitTorrent peers into clusters and we conform to the principle of economic. Experimental results show that we adopt PCS is based on biased neighbor selection and DQA to solve cross-ISP traffic, not only restrain the redundancy but users can obtain a great performance, and decrease the file download time significantly. We have concluded that reforming the performance of BitTorrent should be considering the whole ISP first. Instead of only improving the download rate, we also need to consider the cross-ISP traffic. Thus, ISPs can reduce cross-ISP traffic without sacrificing system performance, and let ISPs and users create a win-win situation. We expected to accommodate the more complex network to our simulator in the future, and make the more complete analysis and consider the factors of influence, which causes the cross-ISP traffic.

Actually, we cannot only rely on decrease redundancy to reduce the cost of equipment in the reality. Because of the lowest redundancy indicates reduce redundancy of only one file. However, a large number of files were shared by P2P systems. We cannot avoid the P2P traffic has been increasing gradually. Thus, ISPs still need to economize on cost of extended equipment by bandwidth limiting. But, the cross-ISP traffic may get out of control if there is no strategy, which reforms the P2P system. We thought an approach combines PCS with bandwidth limiting can reduce cross-ISP traffic and conform to the principle of economic. There are many proposals intend to integrate CDN into P2P system or developing the system such as P4P now, even more let P2P system merge with monitor system. They promote Qos through this equipment manage the files and users, and monitor the traffic. Most importantly, they also the advantages of P2P are preserved, such as scalability, expansibility and arbitrariness. It depends on the consideration of ISPs between the cost and the service. On the other hand, desire of collaboration between ISPs and users, and the incentive that let users adopt the new P2P system also should be considering.

References

1. BitTorrent, `http://www.bittorrent.com/`
2. Napster, `http://www.napster.com/`
3. Gnutella, `http://www.gnutelliums.com/`
4. eMule, `http://www.emule.org/`
5. CacheLogic, `http://www.cachelogic.com/`
6. Aggarwal, V., Feldmann, A., Scheideler, C.: Can ISPs and P2P Users Cooperate for Improved Performance. In: Proceedings of ACM SIGCOMM Computer Communication Review, July 2007, vol. 37(3), pp. 29–40 (2007)
7. Aggarwal, V., Akonjang, O., Feldmann, A.: Improving User and ISP Experience through ISP-aided P2P Locality. In: Proceedings of IEEE Conference on Computer Communications Workshops, INFOCOM 2008, April 13-18, pp. 1–6 (2008)
8. Bharambe, A.R., Herley, C., Padmanabhan, V.N.: Analyzing and improving a BitTorrent network's performance mechanisms. In: Proceedings of the 25th IEEE International Conference on Computer Communication, INFOCOM 2006, April 2006, pp. 1–12 (2006)
9. Bindal, R., Cao, P., Chan, W., Medved, J., Suwala, G., Bates, T., Zhang, A.: Improving traffic locality in BitTorrent via biased neighbor selection. In: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS), July 2006, p. 66 (2006)
10. Burger, M.D., Kielmann, T.: MOB: zero-configuration high-throughputmulticasting for grid applications. In: Proceedings of the 16th international symposium on High performance distributed computing, Monterey, California, USA, June 25-29, pp. 159–168 (2007)

11. Cohen, B.: Incentives Build Robustness in BitTorrent. In: Proceedings of First Workshop on Economics of Peer-to-Peer Systems (2003)
12. Choffnes, D.R., Bustamante, F.E.: Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems. In: Proceedings of the ACM SIGCOMM 2008 conference on Data communication, Seattle, WA, USA, pp. 363–374 (2008)
13. Feldman, M., Papadimitriou, C., Chuang, J., Stoica, I.: Free-riding and whitewashing in Peer-to-Peer systems. In: Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems, Portland, Oregon, USA, pp. 228–236 (2004)
14. Gkantsidis, C., Rodriguez, P.R.: Network coding for large scale content distribution. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2005, March 13-17, vol. 4, pp. 2235–2245 (2005)
15. Guo, L., Chen, S., Xiao, Z., Tan, E., Ding, X., Zhang, X.: Measurements, Analysis, and Modeling of BitTorrent-like Systems. In: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement, Berkeley, CA, p. 4 (2005)
16. Huang, K., Wang, L., Zhang, D., Liu, Y.: Optimizing the BitTorrent performance using an adaptive peer selection strategy. In: Proceedings of Future Generation Computer Systems, July 2008, vol. 24(7), pp. 621–630 (2008)
17. Izal, M., Urvoy-Keller, G., Biersack, E.W., Felber, P., Hamra, A., Garces-Erice, L.: Dissecting bittorrent: Five months in a Torrent's lifetime. In: Proceedings of the 5th Passive and Active Measurements Workshop, France, pp. 1–11 (2004)
18. Izmailov, R., Ganguly, S., Tu, N.: Fast Parallel File Replication in Data Grid. In: Proceedings of Future of Grid Data Environments workshop (GGF-10), Berlin, Germany (March 2004)
19. Legout, A., Urvoy-Keller, G., Michiardi, P.: Rarest first and choke algorithms are enough. In: Proceedings of the 6th ACM SIGCOMM conference on Internet Measurement, Rio de Janeriro, Brazil, pp. 203–216 (2006)
20. Karagiannis, T., Rodriguez, P., Papagiannaki, K.: Should internet service providers fear peer-assisted content distribution. In: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement, Berkeley, CA, p. 6 (2005)
21. Tian, Y., Wu, D., Ng, K.W.: Modeling, analysis and improvement for BitTorrent-like file sharing networks. In: Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM 2006, Barcelona, Spain, April 2006, pp. 1–11 (2006)
22. Xie, H., Yang, Y.R., Krishnamurthy, A., Liu, Y., Silberschatz, A.: P4P: Provider Portal for Applications. In: Proceedings of the ACM SIGCOMM 2008 conference on Data communication, Seattle, WA, USA, pp. 351–362 (2008)