2009 International Conference on Information Technology and Computer Science

A novel peer selection algorithm to reduce BitTorrent-like P2P traffic between networks

Ou-Yang Rong
Dept. of Computer Science
HangZhou DianZi University, China
oyrong@gmail.com

Cao Hui
Shenzhen Telecommunication Bureau, China
gzh_78@sina.com

Abstract

In a peer-to-peer (P2P) file sharing system, the problem of selecting peers to download from, affects greatly the performance, efficiency, and traffic distribution of the whole system. At present, BitTorrent(BT) is most popular P2P software used by large-scale people, and BT traffic is main component in P2P traffic. In order to control the BT-like P2P traffic and reduce their bandwidth occupancy at some key location such as gateway, in this paper, a peer selection algorithm based on file piece convergence degree (FPCD) is brought forward, which takes BT as an example. Applying this algorithm, BT-like P2P traffic between networks can be controlled and reduced heavily, which was testified by modeling and analyzing the peer selection algorithm based on FPCD.

Keywords-Peer selection algorithm; P2P traffic; Networks

1. Introduction

In a peer-to-peer (P2P) file sharing system, data is duplicated among the peers participating in the system. Data source selection during this process means which peers are selected to download from. This problem affects greatly the performance, efficiency, and traffic distribution in the whole P2P system. Many studies on that have been done.

The methodology of machine learning was used in [1] for the construction of good peer selection strategies from past experience. Decision tree learning is used in [1] for rating peers based on low-cost information. The strategies of peer selection can be classified into two categories in [2]: the first category requires global state information of the P2P overlay network to compute an assignment for a request, while the other only needs the state information related to the request, such as the potential serving peers of the request. A genetic-algorithm-based neighbor-selection strategy was proposed in [3] for hybrid P2P networks, which enhances the decision process performed at the tracker for transferring coordination. A scheme which attempts to build a more intelligent overlay network was investigated in the statement of [4], particularly using synthetic network coordinates to select overlay peers that are close in the underlying network. Some super peers were set as tracker in [6], and peers were selected intelligently according to peer capability and the sharing pieces.

At present, P2P traffic has occupied most Internet bandwidth. How to control P2P traffic and reduce the bandwidth occupation in some key location such as gateway is our research goal in this paper. Most of P2P traffic is made of BitTorrent(BT)-like traffic according to [7]. We take BT traffic as an example of P2P traffic.

No considering more about great impact on network bandwidth brought by BT traffic, the peer selection algorithm in BT system mainly took the whole BT system efficiency into account. In this paper, we bring forward a novel peer selection algorithm based on FPCD which can reduce the network bandwidth occupied by BT traffic, without changing the 'Rarest First' and 'Strict Priority' algorithm used by BT system at present which were proved to be efficient.

The remainder of this paper is organized as follows. In Session 2, we describe the peer selection algorithm used by BT system at present. In Session 3, we construct a novel peer selection algorithm based on file piece convergence degree (FPCD) to reduce BT-like P2P traffic between networks. Session 4 makes a model for peer selection algorithm based on FPCD and analyzes it in detail. Finally, we conclude in Session 5.

2. Peer selection algorithm in BT

BT system is made of tracker, downloader and seed. Tracker returns a peers list for each peer to help it finding other peers. A file that will be transferred in BT system is cut into some pieces which have the fixed size (typically, 256k), and each piece is cut into some blocks (typically, 16k). The information about the file and tracker is put into a file whose suffix name is 'torrent'. The peer downloads the 'torrent' file to learn the file information and the network address of tracker, and connects tracker to get peers list. The peer connects the peers in list and downloads the file pieces from them, at the same time, also uploads the file pieces they wanted to them while downloading. The peer will become seed after it have gotten the whole file.

During its lifetime between entering the BT system and leaving, a peer must continue to communicate with other peers using some massages, such as 'handshake' message which is sent to a peer wanted to connect, and 'have' message which shows the peer sending the message has a corresponding piece. The packet format of 'have' message illustrates as Fig.1.

<table>
<tr><td colspan="3">IP header</td></tr>
<tr><td colspan="3">TCP header</td></tr>
<tr><td>0005 (length)</td><td>04 (type)</td><td>Index value of piece</td></tr>
</table>

Fig.1 packet format of 'have' message

During the downloading process, peer *A* which was selected at random from peers in BT system continues to receive the 'have' message from its neighbors. We call the pieces needed to be downloaded as the interested pieces. Peer *A* has a counter which is used to count the numbers of 'have' message for every interested piece. When peer *A* receives a 'have' message, the corresponding counter adds 1. The piece which has the minimum 'have' message number will be chosen to download. That is to say, the corresponding peer will be chosen to download. That is called 'Rarest First'.

During downloading process, peer *A* records the pieces which have been downloading, the blocks of these pieces are requested firstly, this is called 'Strict Priority'. BT keeps sending several requests (usually 5) in pipelining way at the same time. Each request asks for downloading a block [5].

The peer selection algorithm in BT system is determined by piece selection algorithm, namely, 'Rarest First' and 'Strict Priority' algorithm.

3. Peer selection algorithm based on FPCD

3.1. File piece model

We assume that each file is cut into n pieces, and each piece is cut into m blocks. After being cut, the file can be described abstractly as a matrix F . In F , f_i ($1 \le i \le n$) represents the flag of a piece of the file, f_{ij} （$1 \le i \le n, 1 \le j \le m$ ） represents the flag of a block of the file. The coefficient matrix corresponding to matrix F is A:

$$F = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1m} \\ f_{21} & f_{22} & \dots & f_{2m} \\ \dots & \dots & \dots & \dots \\ f_{n1} & f_{n2} & \dots & f_{nm} \end{bmatrix} \quad A = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

$$a_i = \begin{cases} 1 & \text{if } f_i \text{ exist} \\ 0 & \text{if } f_i \text{ not exist} \end{cases} \quad a_{ij} = \begin{cases} 1 & \text{if } f_{ij} \text{ exist} \\ 0 & \text{if } f_{ij} \text{ not exist} \end{cases}$$

Assume the PIECES is a set of the pieces, and the BLOCKS is a set of the blocks. PIECES=$\{f_1, f_2, \dots, f_n\}$, BLOCKS=$\{f_{11}, f_{12}, \dots, f_{1m}, f_{21}, f_{22}, \dots, f_{2m}, \dots, f_{n1}, f_{n2}, \dots, f_{nm}\}$.

Definition 1：The file piece convergence degree (FPCD) among peer *A*, peer *B*, peer *C*, …, peer *X* is

$$s_{ABC\dots X} = \sum_{i=1}^{n} (a_i^{(A)} \parallel a_i^{(B)} \parallel a_i^{(C)} \dots \parallel a_i^{(X)})$$

（'||' is the symbol of 'or' ）

$a_i^{(A)}$ represents the coefficient of piece *i* belonged to peer *A* whose value is 1 or 0. $s_{ABC\dots X}$ means how many different pieces exist among the peers group which made up of peer *A*, peer *B*, peer *C*, …, peer *X* (the peer number in the peers group is more than 2).

Definition 2：The merit degree of peer A to peer B is

$$v_{AB} = s_{AB} - \sum_{i=1}^{n} a_i^{(B)}$$

v_{AB} means how many different pieces peer A has and peer B hasn't.

Definition 3：If $s_{ABC\dots X} = n$ ， then the peers group consisted of peer *A*, peer *B*, peer *C*, …, peer *X* is called as a complete peers group.

It is obvious that any peer in a complete peers group can download the whole file from these peers.

3.2. Peer selection algorithm based on FPCD

We focus on the peers in a BT system which locate in an AS area such as MAN, campus network, enterprise network, etc., and call these peers as interior peers. Those peers which locate out of the AS area are called as exterior peers. According to the border router location for these networks, we define the inside part which include interior peers as internal network, while the outside part as external network. File downloading process in BT system illustrates as Fig.2.

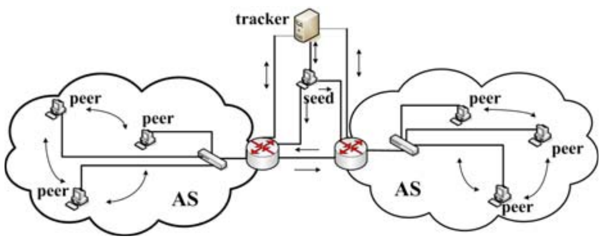

Fig.2 File downloading in BT system

First, we call all peers which are transferring file pieces with peer *A* as the neighbors of peer *A*. For peer *A*, all interested pieces and their 'have' message number are saved in a table. Table 1 is an example table for 'have' message number.

Table 1. Number of 'Have'

piece	number of 'have'
f_1	5
f_2	3
…	…

Table 2. List 1

piece	peer
f_α	peer1
f_β	peer2
…	……

We assume the pieces which are being downloaded are put into List 1, it includes the pieces being downloaded and the corresponding peers. The group consisting of these pieces in list 1 is $\{i_0, i_1, \ldots, i_s\}$, $i_0, i_1, \ldots, i_s \in$ PIECES. Table 2 is an example for List 1.

Table 3. List 2

pieces		peer downloading speed max → min
number of 'have' min ↓ max	f_α	peer1, peer3, ……
	f_β	peer2, peer4, ……
	……	……

We assume the pieces which have not been downloaded are put into List 2. These pieces list with the order of number of 'have' message from minimum to maximum. The peers having these pieces list from maximum to minimum according to downloading speed. The group consisting of the pieces in list 2 is $\{j_0, j_1, \ldots, j_t\}$, $j_0, j_1, \ldots, j_t \in$ PIECES. Table 3 is an example of List 2. The pieces in list 2 are the ones which peer *A* is interested in.

The pieces having been downloaded are put into list 3. The group consisting of these pieces is $\{u_0, u_1, \ldots, u_p\}$, $u_0, u_1, \ldots, u_p \in$ PIECES.

We assume the downloaded blocks which belong to piece i which is being downloaded are put into List11(i). List11(i) $\subset$ BLOCKS. The blocks which are not downloaded and belong to piece i which is being downloaded are put into List12(i). List12(i) $\subset$ BLOCKS.

Finding out all peers which belong to the same physics internal network from neighbor peers list returned from tracker, we call these peers as a local neighbor peer group. We put them into List 4, from maximum to minimum according to their merit degree to peer *A*.

When downloading starts, peer *A* reads out the first piece flag (here is f_α) in List 2 (take Table 3 as an example), which is the rarest piece in all neighbors. If List 2 is empty, then all pieces peer *A* wanted have been downloaded. From List 2, peer *A* finds out the peer (here is peer1) with the highest downloading speed to download f_α. f_α and the corresponding peer1 are written into List 1. Write $f_{\alpha j}$ ($1 \le j \le m$) into List12(α).

The peer *A* sends *s* (default 5 in BT protocol) downloading requests for the blocks corresponding to f_α in pipelining way, and write flag of blocks received from peer1 into List11(α), then take off the downloaded block flag from List12(α). The piece f_α is being downloaded at this time. According to Table 1, peer *A* compute the FPCD among all local neighbor peers. Once they can form a complete peers group, downloading pieces from external network should be stopped at once. Now, read out the first peer from List 4 which has the highest merit degree to peer A. The process of a complete peers group being formed illustrates as Fig.3.

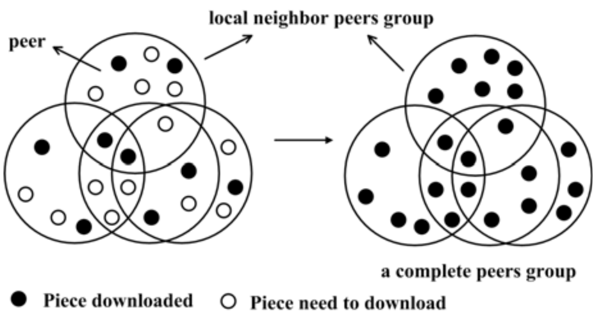

Fig.3 The process of a complete peers group being formed

If local neighbor peers cannot form a complete peers group, then the peer *A* continues to download as usual until a complete peers group appears or downloading end.

4. Algorithm modeling and analyzing

Assume that all peers do not leave the BT system before finishing downloading the file. We consider the arrival of peer accords with Poisson Processes, the arriving rate of peers is λ , $t = 0$ is initial time, and then the number of peers is λt at time t . Among all the peers in this BT system, the ratio of the number of peers locating in internal network is p , the ratio of the number of peers locating in external network is $1-p$. Then the number of peers in internal network is $N_{in} = p\lambda t$, and the number of peers in external network is $N_{out} = (1-p)\lambda t$ at time t .

We assume each peer has reached its maximum downloading connection numbers, D_{dl} , which all peers have the same one. In the downloading connection list, which is with the order of the downloading speed from maximum to minimum, the peers which lie before D_{dl} will be selected as the downloading source. In order to simplify the analyzing process, we just consider the downloading process from external network to internal network. p_1 is the ratio of the number of peers belonging to the internal network in the downloading peers list, and p_2 is the ratio of the number of peers belonging to the external network in downloading peers list. $p_1 + p2 = 1$.

The tracker distributes peers list at random. Assume that each peer in BT network has same p_1 and p_2 , so in the downloading source of downloading connection list of any peer in internal network, the number of peers in internal network is $D_{dl1} = p_1 D_{dl}$, and the number of peers in external network is $D_{dl2} = p_2 D_{dl}$. Assume that each peer in internal network has the same average downloading speed from the peer in internal network, $\overline{b_{dl1}}$, while each peer in internal network has the same average downloading speed from the peer in external network, $\overline{b_{dl2}}$, for every downloading connection.

Before using the peer selection algorithm based on FPCD, each peer (take peer *A* as an example here, which is selected from internal network at random) occupies the export downloading bandwidth, $B_{p-dl2} = \overline{b_{dl2}} D_{dl2} t_p$, t_p is the needed time for peer *A* to finish downloading, $t_p = \frac{256n_1}{\overline{b_{dl1}} D_{dl1}} + \frac{256n_2}{\overline{b_{dl2}} D_{dl2}}$. n_1, n_2 are the number of pieces which are downloaded respectively from internal network and external network by the peer *A*. Thus, the total export downloading bandwidth in a AS area occupied by all peers in internal network is

$$B_{dl2} = B_{p-dl2} N_{in} = \overline{b_{dl2}} p_2 p \lambda t \left(\frac{256n_1}{\overline{b_{dl1}} p_1} + \frac{256n_2}{\overline{b_{dl2}} p_2}\right) \quad (1)$$

We define peer *A* and its local neighbor peers as a local neighbor peer group. The pieces belonging to a local neighbor peer group are all from external network. The arrival of file pieces accords with a Poisson Processes. The arriving rate of file pieces is λ_p , $\lambda_p = (\overline{b_{dl2}} / 256) v$, v is the number of peers in internal network downloading different pieces from external network.

After using the peer selection algorithm based on FPCD, each peer in internal network occupies the export downloading bandwidth, $B_{p-dl2}' = \overline{b_{dl2}} D_{dl2} t_p'$, $t_p' = n / \lambda_p$. Thus, the total export downloading bandwidth occupied by all peers in internal network is

$$B_{dl2}' = B_{p-dl2}' N_{in} = \overline{b_{dl2}} p_2 D_{dl} (n / \lambda_p) p \lambda t$$

(2)

From Equation (1) and (2), we can conclude:

$$k = \frac{B_{dl2}'}{B_{dl2}} = \frac{nD_{dl2} / 256\lambda_p}{\frac{n_1 / \overline{b_{dl1}}}{p_1} + \frac{n_2 / \overline{b_{dl2}}}{1 - p_1}} \quad (3)$$

We assume the size of the file being downloaded is 256M. The average speed of each downloading connection between the peer *A* and other peer in internal network is 50kb/s. The average speed of each downloading connection between the peer A and other peer in external network is 30kb/s. The ratio of the number of peers in internal network is 30%. Before using the peer selection algorithm based on FPCD, the ratio of the numbers of downloaded pieces respectively from internal network and external network is 25%, the number of peers in internal network downloading different pieces from external network is 50, the maximum downloading connection number is 30, p_1 varies from 0.3 to 0.8. According to Equation (3), we can get the relation curve of k and p_1 as Fig. 4.

From Fig.4, we know k <0.4. That is to say, after using the peer selection algorithm based on FPCD, the export downloading bandwidth drops to more than 40%. Along with the increase of p_1, the export downloading bandwidth will drop more.

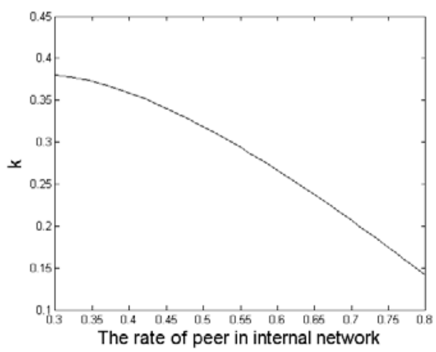

Fig. 4 The relation curve of k and p_1

5. Conclusions

In this paper, we construct a novel peer selection algorithm based on FPCD to reduce BT-like P2P traffic between networks. From modeling and analyzing the peer selection algorithm based on FPCD, we can conclude, after using the peer selection algorithm based on FPCD, BT-like P2P traffic between networks are reduced heavily. In this way, the traffic congestion between networks brought by BT-like P2P downloading will be alleviated greatly.

References

[1] Daniel S. Bernstein, Z.F., Brian Neil Levine, et al., "Adaptive peer selection", *In Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS03),* Berkeley, CA, February 2003.

[2] Ammar, "The effect of peer selection and buffering strategies on the performance of peer-to-peer file sharing systems", *10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, Mascots, 2002, pp. 63-70.

[3] Koo, S.G.M., Lee, et al., "A genetic-algorithm-based neighbor-selection strategy for hybrid peer-to-peer networks", *ICCCN Proceedings 13th International Conference,* 2004, pp. 469-474.

[4] Qureshi, A., "Exploring proximity based peer selection in a BitTorrent-like protocol", *MIT 6.824 student project,* May7, 2004.

[5] Cohen B., "Incentives build robustness in bittorrent", *http://bitconjurer.org/BitTorrent/bittorrentecon.pdf,* May 2003.

[6]Rong, L.B., I., "BitTorrent in a dynamic resource adapting peer-to-peer network", *First International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution,* 2005.

[7]CacheLogic, "The true picture of peer-to-peer filesharing", *http://www.cachelogic.com/press/CacheLogic_Press_and_Analyst_Presentation_July2004.pdf,* July 2004.