This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Improving the Download Time of BitTorrent-like Systems

Chi-Jen Wu, Cheng-Ying Li, and Jan-Ming Ho
Institute of Information Science
Academia Sinica, Taiwan
{cjwu, cyli, hoho}@iis.sinica.edu.tw

Abstract— **The content distribution techniques have recently started embracing peer-to-peer system as an alternative to the client-server architecture, such as BitTorrent system. BitTorrent system offers a scale mechanism for distributing a large volume of data to a set of peers over the Internet, but it is not designed for minimizing the time taken for all peers to receive the file. As a result, the peers of BitTorrent system may suffer a long download time, specifically the narrow-band peers. In this paper, in order to reduce the download time of BitTorrent, we propose a weighty piece selection strategy instead of the local rarest first strategy in BitTorrent. The proposed strategy is based on the greedy concept that a peer assigns each missing piece a weight according to total number of neighbor's downloaded pieces. The peer selects the missing piece with the highest priority for next download. This strategy can speed up the cooperation between heterogeneous peers while making the BitTorrent more efficient in terms of the average download time and the total elapsed time. The simulation results show that weighty piece selection strategy can improve more than 15% average download time and reduce in average 60% total elapsed time than the BitTorrent system.**

I. Introduction

In recent years, the content distribution techniques have started embracing peer-to-peer (P2P) system as alternative to the client-server architecture such as the BitTorrent system [1], a yet popular and successful P2P content distribution system. A recent research [2] shows the BitTorrent system accounts for 35% of all the traffic on the Internet. Compared with the traditional client-server architecture, BitTorrent system offers a scalable manner of delivering a large volume of data from an origin server to a large set of clients. However, the BitTorrent system is fundamentally different from all previous P2P application that relies on the participators sharing contents in a P2P system; it creates a new P2P content distribution mechanism for large scale Internet users.

The BitTorrent system can tolerate the fluid environment of the Internet, in a better scalable manner than other application-level multicasting tools (NICE [3], Narada [4], although, these mechanisms are extreme scalability, these researches focus on building an efficient and low end-to-end latency overlay). Actuality, the BitTorrent system is wide used and receives a lot of attentions from researchers of the networking and distribution system area [5-10]. Many research papers [5-7] have been conducted to study the performance of BitTorrent-like systems. They show that BitTorrent system is both efficient and scalable, even BitTorrent is without central coordination and scheduling. The lack of centralized planning and scheduling makes BitTorrent easy to develop but it has some disadvantages; it does not consider the performance of overall peers, such as average download time and the total elapsed time of all peers. Let us consider the scenario that an enterprise of online game, ex Blizzard Entertainment [11] needs to upgrade the software of end users. The average download time and the total elapsed time of all users will become the important key metrics for all users and the business company. However, based on the fluid properties of distribution systems, to minimize the client download time in a large distribution system is very difficult. This is especially the case in practical systems that can not rely on a central scheduler and, instead, allow clients to make local decisions. Moreover, when the number of clients increases, the scheduling problem will become more difficult. For these reasons, how to minimize the average download time and the total elapsed time of all peers in the BitTorrent system is an interesting problem.

In this paper, we propose a weighty piece selection strategy that performs significant improvement of BitTorrent-like system in terms of the average download time and the total elapsed time of all peers. Our weighty piece selection strategy is based on the greedy concept that each missing piece is assigned a weight according to the total number of neighbor's download pieces. When a peer wants to select a piece to download, it shall select the missing piece with highest weight. On the other hand, a new peer or the peer that has few pieces shall select and download the piece that desired and searched by its neighbors with more pieces. The weighty piece selection strategy is also compared with the local rarest first strategy of the BitTorrent system. Based on the extensive simulations, the proposed strategy can save the approximate 60% total elapsed time of the BitTorrent system and reduce more than 15% average download time of overall system. We rely on simulations since it is difficult to capture the piece selection strategy or other relevant mechanism in the BitTorrent system in an analysis model. The weighty piece selection strategy can be also viewed as a technique for BitTorrent peers to share contents more cooperatively, thus the technique has the potential to function as Internet-wide software upgrade mechanism for a global enterprise.

The rest of this paper is structured as follows: in the next section, we describe and review previous work on BitTorrent-like systems in the section II. Section III describe the details of weighty piece selection strategy, and compare it to the

1-4244-0353-7/07/$25.00 ©2007 IEEE

local rarest first strategy of the BitTorrent system. Section IV explains the simulation methodology used in this study and presents the performance results of the simulation study. Section V concludes this paper with a summery of the main research results from this study.

II. Related Work

In this section, we will describe a brief summarize of related studies about the BitTorrent-like systems. There have many research results of the BitTorrent system include analysis, modeling and performance improvement of the BitTorrent-like systems. The extensive measurements and traffic analysis on BitTorrent systems have been conducted in the study [10]. The authors measure and analyze a five month period of a thousands peers BitTorrent system by investigating the tracker log of the popular torrent. Their results serve as the empirical evidence that the BitTorrent system is highly efficient in the Internet, but this study does not try to improve the performance of the BitTorrent system. Yang et al. [5] study the service capacity of BitTorrent-like systems by using branching processes to model BitTorrent systems. Qiu and Srikant [6] devise a fluid model to provide a good starting point for analyzing the BitTorrent-like system's performance behavior. This model characterizes the overall performance of BitTorrent-like systems and analyzes the effectiveness of the TFT peer selection policy. And in [7], a mathematical model for BitTorrent system is presented. The modeling results find that the online peer distribution follows a U-shaped curve. The simulation approach of BitTorrent systems is studied in [8], their results show that BitTorrent system is effective in utilizing the available uploading bandwidth, the authors also proposed a enhance version of the TFT mechanism to eliminate the unfairness problem of the BitTorrent system.

Other previous efforts attempted to improve upon the BitTorrent-like system. Slurpie [12] is designed to reduce client download time for large files. The purpose of Slurpie is similar to ours, but the algorithms of Slurpie are much more complex than the ones of BitTorrent system, and need to know the number of peers in a Slurpie system. Though simulation results of Slurpie in [12] is outperform BitTorrent system, the cases of flash crows and for a large number of peers are unknown. In [13], Gkantsidis and Rodriguez propose to use network coding to improve performances of the BitTorrent-like system. With the network coding mechanism, each peer in the network is able to generate and transmit encoded pieces of data to enhance the diversity of piece. Their simulation results show an improvement of 2-3 times over un-encoded version of BitTorrent, but the actual performance gains over BitTorrent system in real Internet is also unknown. An enhanced version of BitTorrent is proposed in [9], the authors consider the traffic locality problem of BitTorrent. The proposed peer selection strategy can reduce significant network traffic between ISPs.

Compared with the above efforts, this work makes the following contributions. First, we devise a new piece selection strategy for the BitTorrent system that diminishes the total elapsed time of overall system and also reduces the average download time. We only compare our algorithm with the BitTorrent system, due to the optimal download time of a dynamic BitTorrent system is hard to estimate. However, we will try to analyze the optimal download time of a BitTorrent system in the future work. Second, the weighty piece selection strategy is very easy and simple to be implemented into a BitTorrent system. Our algorithm use the local view information of a network as same as the LRF strategy, a peer picks a suitable piece according to the file piece distribution among its neighboring peers. Compared with original BitTorrent system, our algorithm reduces the download time of BitTorrent system and it is more suitable for a global business company to distribute content (software) to customers.

III. The Piece Selection Strategies

The piece selection strategy in BitTorrent system describes the selection of the next piece be chosen to transfer. When a new peer first joins the BitTorrent network by a .torrent file, the peer first randomly choices four pieces to download in the initial phase. After the initial phase, the peer should use one of above strategies to pick the next download piece. The decisions of piece selection strategy have direct impact onto the performance of the BitTorrent system as well as the peer selection mechanism, but in this paper we focus on the piece selection algorithm for BitTorrent. Following, the system model, our assumptions and two possible algorithms are described and discussed: local rarest first piece strategy, and the proposed weighty piece selection strategy.

A. System Model and Assumptions

In this section the system model and our assumptions are introduced. In a BitTorrent system architecture, there is no dependence costly pre-provisioned infrastructure, making it favorable as an economical and quickly deployable alternative for content distribution applications. But the current BitTorrent protocol is not designed for minimizing the download time of overall system, so that we try to enhance the performance of the BitTorrent system in this aspect. We assume a large number of users interested in some content and the content provider (the seed that never leaves) in a BitTorrent system tries to maximize the all users benefit (try to minimize the average download time of overall users). However, the users are strategic, i.e., a user leaves the system as soon as possible after completely receives the file.

We assume a heterogeneous network environment in which all peer are equipped with an asymmetric access link with a different upload and download bandwidth, such as ADSL or Cable link (the details of network bandwidth will be described in Section 5), so that the download time of a file piece is the size of a piece divided by the associated upload connection's bandwidth. Moreover, the core network is assumed to have infinite capacity, it is reasonable due to the empirically the capacity of the core network is under-utilized. In addition, we assume all peers to use the same piece selection strategy, either using the local rarest first strategy or using our weighty piece selection strategy. And other mechanisms, which include

TABLE I
NOTATIONS

Notation	Definition
M_i	the set of pieces that peer i missing
D_i	the set of pieces that peer i downloaded
$N(i)$	the set of neighbors of peer i
$L(D_i)$	the size of D_i
B	the set of overall pieces
α	the parameter of our algorithm

TFT strategy and joining protocol, are based on the BitTorrent system.

The system parameters and notations are summarized in Table 1. Note that they are defined with respect to the distribution of a specific file. We denote by B the set if overall pieces in the file being distributed and by M_i and D_i the set of pieces that peer i has already downloaded and is still missing, respectively (with $B = M_i \cup D_i$ and $M_i \cap D_i = null$). Similarly, $M_i \cap M_n \neq null$ is donated that the peer i and n both are missing a piece m, and they try to find and download a piece m, contrariously one of them had a piece m.

B. Local Rarest First Strategy

The BitTorrent system employs the Local Rarest First strategy for choosing new pieces to download from peers. The main purpose of local rarest first strategy is to overcome the last piece problem [13] by favoring rare pieces. A BitTorrent peer using this strategy selects the requesting piece which has the smallest number of its neighbors. This mechanism results typically in an evenly spread number of sharing peers for all pieces of the file. For applying the local rarest first strategy, each BitTorrent peer maintains the number of copies in its neighbors for each piece. We use this information to define the rarest pieces. The requesting peer r selects the rarest piece $m \in (D_n \cap M_r)$ among those that it misses and one of its neighbor, receiving peer n, held. The rarest piece is computed form the number of each piece that held by the neighbors of requesting peer r. More precisely, we use the following computing function for the local rarest first strategy:

$$\forall m \in (D_n \cap M_r)$$

$$F(m) = \begin{cases} \sum\limits_{i \in N(r)} L(D_i)^{\alpha=0}, & \text{if } m \in M_i \\ 0, & \text{otherwise.} \end{cases}$$

choosing a piece $\hat{m} \ni F(m)$ is max.

This function shows that computing model of the local rarest first strategy for a peer r. For each piece m, a peer, r calculates the number of neighboring peers that request the piece, and it chooses a piece with max value for next download. We let parameter $\alpha = 0$, it means that all missing pieces in local rarest first strategy are equal for all neighboring peer. So that the peer, r elects a missing piece based on the number of the neighboring peers which also require this piece.

This strategy is expected to maximize the number of the rarest piece in the BitTorrent system. To maintain the diversity of pieces is very important in BitTorrent-like systems, because the diversity of pieces is responsible for the file availability and the content lifetime of the BitTorrent network. Recall that our scenario, the content provider of Software Company (the seed) should never leave, so that the file availability will be not a problem in here. Next we will describe our weighty piece selection strategy.

C. Weighty Piece Selection Strategy

In this section, we describe our weighty piece selection mechanism for BitTorrent-like systems. The function of weighty piece selection is similar the one of local rarest first strategy in the BitTorrent system, they both are for choosing new pieces to download from peers. The goal of weighty piece selection is to reduce the average download time and total elapsed time of BitTorrent-like system. Logically, a requesting peer r selects the weightiest piece $m \in (D_n \cap M_r)$ among those that it misses and one of its neighbor, receiving peer n, held. The weight is computed by the sum of $L(D)$ of all neighbors which also lack this piece.

Without loss of generality, we show the computing function of the weighty piece selection mechanism in following:

$$\forall m \in (D_n \cap M_r)$$

$$F(m) = \begin{cases} \sum\limits_{i \in N(r)} L(D_i)^{\alpha=1}, & \text{if } m \in M_i \\ 0, & \text{otherwise.} \end{cases}$$

choosing a piece $\hat{m} \ni F(m)$ is max.

This function abstracts that computing model of the weighty piece selection strategy for a peer r. For each piece m, a peer, r calculates the sum of $L(D)$ of neighboring peers that request the piece, and it chooses the piece with max value for next download. We let $\alpha = 1$, it means that a missing piece is assigned a weight according the sum of $L(D)$ of neighboring peers which also lack this piece. A peer shall select and download the missing piece that desired by the neighbors with more pieces. It implies that the plentiful peer can find and download missing pieces from its neighboring peers that have few pieces with high probability. In addition, the piece inter-arrival time will decrease with the number of the downloaded pieces increasing. In addition, the peers with few pieces can have great chances to exchange the missing pieces with the plentiful peers. So that our weighty piece selection strategy can efficiently diminish the average download time and total elapsed time of BitTorrent-like systems. Note that the performance of the weighty piece selection strategy is not optimal, and how to minimize the average download time of a content distribution system without central coordination and scheduling is an open problem [15].

Compared with the local rarest first strategy, it prevents last piece problem and increases the file availability of BitTorrent systems. For the user satisfaction, this two metrics are not most important in the software update scenario, the average download time and total elapsed time should be more important in here.

IV. Performance Evaluation

To understand the interaction of the weighty piece selection with other BitTorrent mechanisms, we built a discrete-event simulator to simulate the downloading of a large file. Our simulator is based on the paper [8] that is the first paper that studied the performance of the BitTorrent-like system by simulation. In this section, we present the details of the simulator and describe the performance metrics we focus on in our evaluation. The network model associates a downloadlink and an uplink bandwidth for each peer, which allows modeling asymmetric access networks. The table 2 summarizes the distribution of peer bandwidths. This peer bandwidth distribution reported for Gnutella clients [14]. We only simulate the heterogeneous network, due to the homogeneous network is not existing in the actual Internet. We use the following setting in our experiments, although we do vary these setting in some experiments as noted in later:

- File size: 204800 KB = 200 MB (800 pieces of 256 KB each)
- Number of initial seeds: 1 (the server, which stays on throughout the duration of the simulation)
- Uplink bandwidth of seed: 6000 kbps
- Number of downloading peers that join the system: 500 ∼ 5000
- Join/leave ratio: 5 peers/per sec., and peers depart as soon as they finish downloading
- Number of concurrent upload transfers: 5 (includes the optimistically unchoked connection)

We evaluate the effectiveness of the weighty piece selection strategy (WLRF) in terms of the following metrics: (a) average download time, (b) total elapsed download time, (c) piece inter-arrival time. All results are compared with local rarest first strategy (LRF).

A. Experiment Results

In the section, we analyze the efficiency of the weighty piece selection (WLRF) strategy by comparing it against local rarest first (LRF) strategy. We evaluate the performance of the WLRF strategy with increasing network size. We vary the number of download peers that join the system from 500 to 5000. All simulations run 20 times and are with the join ratio 5 peers/per sec. (i.e. for 500 peers, all peers join during a 100 seconds period.). We compare WLRF strategy and LRF strategy in terms of their mean download time and mean total elapsed download time. In the software update scenario, this two metrics are very important for user satisfaction. More specifically, we define a compared metric that dedicated *improvement ratio = (download time of LRF - download time of WLRF) / download time of LRF*. Note that we only change the piece selection strategy of BitTorrent system, other mechanisms of BitTorrent system are not modified.

TABLE II
Bandwidth distribution of peers

Downloadlink	Uplink	Fraction
784kbps	128kbps	20%
1500kbps	384kbps	40%
3000kbps	1000kbps	25%
10000kbps	5000kbps	15%

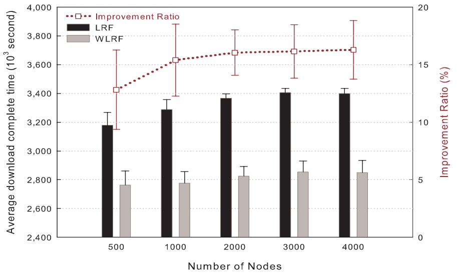

Fig. 1. The average download time

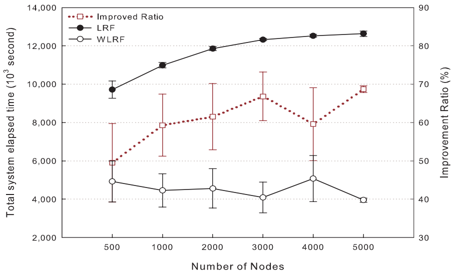

Fig. 2. The total elapsed download time

Fig. 1 shows the improvement ratio of mean download time between WLRF strategy and LRF strategy in the heterogeneous network when the number of peers from 500 to 4000. The improvement ratio does not vary much with the number of peers for a give networks. When the size of network increases, the improvement ratio also goes up. Compared with the LRF strategy, the WLRF strategy has more than 15% improvement in mean download time of BitTorrent system. Fig. 2 shows the improvement ratio of mean total elapsed download time between WLRF strategy and LRF strategy when the number of peers from 500 to 5000. We show that our WLRF strategy can perform outstanding improvement ratio in total elapsed download time, the improvement ratio is as high as 70%. Surprisingly, the total elapsed time of our WLRF strategy is only 1/3 time of the LRF strategy.

Fig. 3 shows the CDFs of the download time for the two strategies in the case of a 4,000 peers BitTorrent system. The top is the CDF of LRF strategy and bottom is the CDF of the WLRF strategy. The WLRF strategy provides a much faster file transmission than the LRF strategy. For clarity,

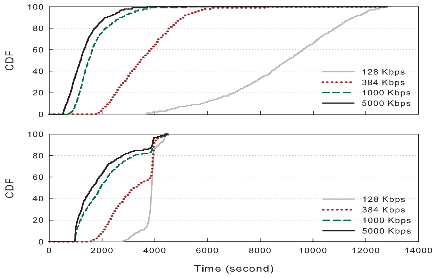

Fig. 3. The CDF of WLRF and LRF strategy

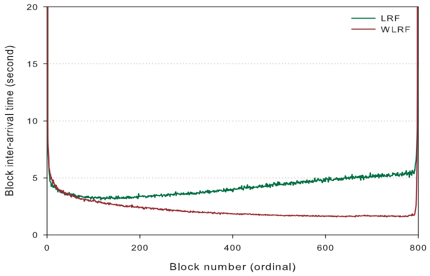

Fig. 4. The piece inter-arrival time

the results are presented that the more than 80% percentile download peers using the WLRF strategy finished download jobs at 4,000 seconds.We can observe that the distribution of download time for the narrow-band peers using LRF strategy specifically is a long tail distribution, it is not a good thing for software update scenarios. If the download time of file is too long, the impression of users will be disappointing for the software company.

Next we plot in Fig. 4 the piece inter-arrival time between pieces in the case of a 4,000 peers BitTorrent system. This is the time between the receipts of consecutive distinct pieces, averaged across all peers. We plot this for both the WLRF and LRF strategies. The sharp downswing in curve corresponding to the WLRF strategy clearly indicates that efficiency of our WLRF strategy. The piece inter-arrival time decreases with the number of downloaded pieces increasing. It is the characteristic of our WLRF strategy.

In summary, these results demonstrate that use of our WLRF strategy for BitTorrent-like system has a significant improvement of the download time in the software update scenario with a large scale download peers. Our WLRF strategy doe not minimize both the mean download time and the total elapsed time of BitTorrent system, but it provides an easy development mechanism to diminish these two metrics.

V. Conclusions and Future work

Large scale and fast distribution of software updates to millions of Internet users is becoming a critical task in today Internet. It seems that the BitTorrent-like system is very suitable for software updates service in the Internet environment, due to the BitTorrent-like systems is easy development and can tolerate properties of the fluid Internet environment. So that, making the BitTorrent-like systems more efficient for software updates services is an interesting problem. This paper proposed a new piece selection strategy for BitTorrent-like systems to address this problem. Based on the simulation results that we have performed, the proposed weighty piece selection that improves the significant performance of BitTorrent-like system in terms of the mean download time and total elapsed time. These two metrics are very important for user satisfaction. The weighty piece selection strategy is also very easy and simple to be implemented into a BitTorrent-like system. Compared with local rarest first strategy of BitTorrent system, this new piece selection strategy is suitable for software updates service.

To minimize the average download time in a large distribution system is still an open problem. In particular, the piece and peer selection strategies impact the peers and global throughput of the system, we will try to rethink the design of BitTorrent-like systems and to investigate a realizable algorithm that can achieve the optimal average download time and the optimal total elapsed time of BitTorrent-like systems in the future work.

References

[1] B. Cohen. Incentives build robustness in BitTorrent. *in Proceedings of first ACM workshop on Economics of Peer-to-Peer System*, Berkeley, USA, June 2003.

[2] T. Karagiannis, A. Broido, M. Faloutsos, Kc claffy. Transport Layer Identification of P2P Traffic. *in Proceedings of ACM IMC*, Taormina, Sicily, Italy, October 25-27, 2004.

[3] Suman Banerjee, Bobby Bhattacharjee, Christopher Kommareddy. Scalable Application Layer Multicast. *in Proceedings of ACM Sigcomm*, Pittsburgh, Pennsylvania, August 2002.

[4] Yang-Hua Chu, Sanjay Rao, and Hui Zhang. A case for end system multicast. *in Proceedings of ACM Sigmetrics*, Santa Clara, CA, 2000.

[5] X. Yang and G. de Veciana. Service Capacity of Peer to Peer Networks. *in Proceedings of IEEE INFOCOM*, Apr, 2004.

[6] D. Qiu and R. Srikant Modeling and performance analysis of Bittorrent-like peer-to-peer networks. *in Proceedings of ACM Sigcomm*, Portland, OR, USA, 2004.

[7] Ye Tian, Di Wu, and Kam-Wing Ng. Modeling, Analysis and Improvement for BitTorrent-Like File Sharing Networks. *in Proceedings of IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[8] A. Bharambe, C. Herley, and VN Padmanabhan. Analyzing and Improving BitTorrent Performance. *in Proceedings of IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[9] Ruchir Bindal, Pei Cao, William Chan, Jan Medved, George Suwala, Tony Bates, Amy Zhang. Improving Traffic Locality in BitTorrent via Biased Neighbor Selection. *in Proceedings of IEEE ICDCS*, Lisboa, Portugal, July 2006.

[10] M. Izal, G. Urvoy-Keller, E.W. Biersack, P.A. Felber, A. Al Hamra, and L. Garces-Erice. Dissecting BitTorrent: Five months in a torrent's lifetime. *in Proceedings of the 5th Passive and Active Measurement Workshop*, Apr. 2004.

[11] Blizzard Entertainment. http://www.blizzard.com/.

[12] R. Sherwood, R. Braud, and B. Bhattacharjee. Slurpie: A Cooperative Bulk Data Transfer Protocol. *in Proceedings of IEEE INFOCOM*, Apr. 2004.

[13] C. Gkantsidis, P. Rodriguez. Network Coding for Large Scale Content Distribution. *in Proceedings of IEEE INFOCOM*, Miami, March 2005.

[14] Stefan Saroiu, P. Krishna Gummadi, Steven D. A Measurement Study of Peer-to-Peer File Sharing Systems. *in Proceedings of ACM MMCN*, Jan 2002.

[15] Gang Wu, Tzi-cker Chiueh. How Efficient is BitTorrent?. *in Proceedings of ACM MMCN*, Jan 2006.