Analysis of Random Time-Based Switching for File Sharing in Peer-to-Peer Networks

Keqin Li
Department of Computer Science
State University of New York
New Paltz, New York 12561, USA
Email: lik@newpaltz.edu

***Abstract*—The expected file download time of the randomized time-based switching algorithm for peer selection and file downloading in a peer-to-peer (P2P) network is still unknown. The main contribution of this paper is to analyze the expected file download time of the time-based switching algorithm for file sharing in P2P networks when the service capacity of a source peer is totally correlated over time, namely, the service capacities of a source peer in different time slots are a fixed value. A recurrence relation is developed to characterize the expected file download time of the time-based switching algorithm. Is is proved that for two or more heterogeneous source peers and sufficiently large file size, the expected file download time of the time-based switching algorithm is less than and can be arbitrarily less than the expected download time of the chunk-based switching algorithm and the expected download time of the permanent connection algorithm. It is shown that the expected file download time of the time-based switching algorithm is in the range of the file size divided by the harmonic mean of service capacities and the file size divided by the arithmetic mean of service capacities. Numerical examples and data are presented to demonstrate our analytical results.**

***Keywords*-chunk-based switching; download time; file sharing; peer-to-peer network; peer selection; time-based switching.**

I. Introduction

A peer-to-peer (P2P) network employs diverse connectivity among participating peers and the combined resources of participants to provide various services [3]. A unique feature of P2P networks is that all peers contribute resources, including storage space, computing power, and communication bandwidth. In addition to the advantage of scalability, the distributed nature of a P2P network also increases the robustness of the network and the capability of fault tolerance in case of peer failures by replicating data over multiple peers.

File sharing using application layer protocols such as BitTorrent is the most popular application of P2P networks, in addition to many other applications such as telephony, multimedia (audio, video, radio) streaming, discussion forums, instant messaging and online chat, and software publication and distribution. The increasing popularity of the MP3 music format in the late 1990s led to Napster and other software designed to aid in the sharing of electronic files. Current popular P2P networks/protocols include Ares Galaxy, eDonkey, Gnutella, and Kazaa [2].

Performance measurement, modeling, analysis, and optimization of file sharing in P2P networks has been a very active research area in the last few years. Research has been conducted at three different levels, i.e., system level, peer group level, and individual peer level. At the system level, research is focused on establishing models of P2P networks such as queueing models [11], [17] and fluid models [10], so that overall system characterizations such as system throughput and average file download time can be obtained. At the peer group level, research is focused on distributing a file from a set of source peers to a set of

978-1-4244-6534-7/10/$26.00 ©2010 IEEE

user peers so that the overall distribution time is minimized [12], [13], [14], [15], [16], [18]. At the individual peer level, research is focused on analyzing and minimizing the file download time of a single peer [8].

It is clear that the vast majority of file downloads are performed by individual users. Therefore, P2P network performance optimization from a single peer's point of view becomes an interesting and important issue. File download includes two parts, namely, file searching and file transfer. Since file searching takes a very small portion of file download time, by file download time, we mean file transfer time. In a P2P network with heterogeneous source peers, after searching and determining the source peers of a file of interest, the major problem for an individual user peer is the peer selection problem, namely, switching among source peers and finally settling on one, while keeping the total time of probing and downloading to a minimum [6]. The problem is called the server selection problem in WWW client-server applications [7], [9]. The peer selection problem is also studied in the context of free-market economy with additional consideration of cost of download [4], [5].

A number of randomized peer selection and file downloading algorithms were proposed in [8], including the permanent connection algorithm, the chunk-based switching algorithm, and the time-based switching (also called periodic switching) algorithm. The expected file download time of the first two algorithms is easy to obtain, namely, the file size divided by the harmonic mean of service capacities. However, the expected file download time of the time-based switching algorithm is still unknown. In [8], it is mentioned that if the service capacities of a source peer in different time periods are lightly correlated or almost independent, then the expected file download time of the time-based switching algorithm is approximately the file size divided by the harmonic mean of expected service capacities. However, the assumption of almost independent service capacities in different time periods is unrealistic. Therefore, finding the expected file download time of the time-based switching algorithm for highly or totally correlated service capacities becomes an interesting and important problem. Such analysis is necessary when the time-based switching algorithm is to be compared with the permanent connection algorithm and the chunk-based switching algorithm.

The main contribution of this paper is to analyze the expected file download time of the time-based switching algorithm for file sharing in P2P networks when the service capacity of a source peer is totally correlated over time, namely, the service capacities of a source peer in different time slots are a fixed value. We give a recurrence relation to characterize the expected file download time of the time-based switching algorithm. We show that for a fixed group of source peers and a given length of time period, the expected file download time of the time-based switching algorithm is a piecewise linear function of file size.

Furthermore, we compare the performance of the time-based switching algorithm with the permanent connection algorithm and the chunk-based switching algorithm. We prove that for two or more heterogeneous source peers and sufficiently large file size, the expected file download time of the time-based switching algorithm is less than the expected download time of the chunk-based switching algorithm and the expected download time of the permanent connection algorithm. More specifically, for a fixed file size, the expected file download time of the time-based switching algorithm is a nondecreasing function of the length of time period. When the length of time period is sufficiently large, the time-based switching algorithm is identical to the chunk-based switching algorithm and the permanent connection algorithm, namely, the expected file download time of the time-based switching algorithm is the file size divided by the harmonic mean of service capacities. When the length of time period approaches zero, the expected file download time of the time-based switching algorithm approaches the file size divided by the arithmetic mean of service capacities. The performance of the random permanent connection algorithm and the random chunk-based switching algorithm can be arbitrarily worse than the performance of the random time-based switching algorithm. We also present numerical examples and data to demonstrate our analytical results.

II. Algorithms and Analysis

Assume that n peers $1, 2, ..., n$ have been identified as source peers of a file of interest, such that any part of the file can be downloaded from any of these n source peers. We assume that the service capacity C_i of source peer i remains the same throughout the session of transferring a file. However, the service capacity C_i of source peer i may be different during different sessions in transferring different files. Hence, the service capacity C_i is unknown when a file is to be downloaded. The n source peers are homogeneous if $C_1 = C_2 = \cdots = C_n$.

We use S to represent the size as well as the name of a file. Let $T_i(S)$ be the download time of a file of size S from source peer i. It is clear that $T_i(S) = S/C_i$. Let $T_A(S)$ denote the expected download time for a file of size S by using a randomized algorithm A from n source peers with service capacities $C_1, C_2, ..., C_n$.

A. Random Permanent Connection

The random *permanent connection* (PC) algorithm works as follows. To download a file, a source peer i is chosen randomly from n source peers with a uniform distribution, that is, each source peer is selected with equal probability $1/n$. Proposition 1 states that the expected file download time of the random permanent connection algorithm is the file size divided by the harmonic mean of service capacities $C_1, C_2, ..., C_n$.

Proposition 1: The expected file download time $T_{\rm PC}(S)$ of the random permanent connection algorithm is

$$T_{\rm PC}(S) = \frac{1}{n}\sum_{i=1}^{n}\frac{S}{C_i} = S\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right).$$

Proof. If source peer i is chosen, the file download time is $T_{\rm PC}(S) = T_i(S) = S/C_i$, which happens with probability $1/n$. Thus, the expected file download time of the random permanent connection algorithm is

$$\begin{aligned} T_{\rm PC}(S) &= \frac{1}{n}\sum_{i=1}^{n}T_i(S) \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{S}{C_i} \\ &= S\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right). \end{aligned}$$

This proves the result. ∎

B. Random Chunk-Based Switching

In a chunk-based switching algorithm, a file to be downloaded is divided into chunks of size S^*, where S^* is a network-wide parameter agreed by and acceptable to all service and user peers. Without loss of generality, it is assumed that S can be divided by S^* and $m = S/S^*$ is the number of chunks, such that the chunks are numbered by $1, 2, ..., m$. Given a file of size S and n source peers, a download schedule specifies a source peer for each chunk.

In the random *chunk-based switching* (CBS) algorithm, a source peer is randomly and uniformly chosen from the n source peers for each chunk. Proposition 2 states that the random chunk-based switching algorithm has the same expected file download time as that of the random permanent connection algorithm.

Proposition 2: The expected download time $T_{\rm CBS}(S)$ of the random chunk-based switching algorithm is

$$T_{\rm CBS}(S) = \frac{1}{n}\sum_{i=1}^{n}\frac{S}{C_i} = S\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right).$$

Proof. It is clear that the expected time to download one chunk is $T_{\rm PC}(S^*)$. Since there are m chunks, we get

$$\begin{aligned} T_{\rm CBS}(S) &= mT_{\rm PC}(S^*) \\ &= mS^*\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right) \\ &= S\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right). \end{aligned}$$

This proves the result. ∎

C. Random Time-Based Switching

The random *time-based switching* (TBS) algorithm divides time into periods of equal length τ. During each time period, a fragment of a file is downloaded from a source peer which is randomly chosen from the n source peers with equal probability. Due to different service capacities of the source peers, the sizes of the fragments downloaded during different time slots may be different.

It seems that the uniform distribution in selecting a source peer in each time period does not make the TBS algorithm to be different from the PC and CBS algorithms. However, it turns out that the TBS algorithm performs more efficiently than the PC and CBS algorithms.

Before making such a comparison, we need to know the expected file download time of the time-based switching algorithm. The following theorem gives a recurrence relation for $T_{\rm TBS}(S)$.

Theorem 1: The expected file download time $T_{\rm TBS}(S)$ of the random time-based switching algorithm is characterized by the following recurrence relation:

$$T_{\rm TBS}(S) = \frac{1}{n}\sum_{i=1}^{n}\left(S > \tau C_i\ ?\ \tau + T_{\rm TBS}(S-\tau C_i)\ :\ \frac{S}{C_i}\right).$$

(Note: The value of an expression $c\ ?\ a : b$ is a if condition c is satisfied and b otherwise.)

Proof. Consider the first period τ. Assume that source peer i is chosen. This happens with probability $1/n$. If $S > \tau C_i$, a fragment of a file of size τC_i is downloaded from source peer i, and the expected download time for the rest of the file of size $S - \tau C_i$ is $T_{\rm TBS}(S-\tau C_i)$, and the total download time is $\tau + T_{\rm TBS}(S-\tau C_i)$. If $S \le \tau C_i$, a file is downloaded in one period of time with length S/C_i. Summarizing the above discussion, the expected file download time $T_{\rm TBS}(S)$ of the random time-based switching algorithm is

$$T_{\rm TBS}(S) = \frac{1}{n}\sum_{i=1}^{n}\left(S > \tau C_i\ ?\ \tau + T_{\rm TBS}(S-\tau C_i)\ :\ \frac{S}{C_i}\right).$$

This proves the theorem. ∎

It is always interesting to solve a recurrence relation. Unfortunately, it seems unlikely that there exists a closed form expression for $T_{\rm TBS}(S)$. Although there is no closed form expression of $T_{\rm TBS}(S)$, we at least know the following.

Theorem 2: The expected file download time $T_{\rm TBS}(S)$ is a piecewise linear function of S in the subintervals of $(0,\infty)$ cut by $\cup_{i=1}^{n}\{\tau C_i, 2\tau C_i, 3\tau C_i, ...\}$.

Proof. Without loss of generality, we assume that $C_1 \le C_2 \le \cdots \le C_n$. Let the set

$$\bigcup_{i=1}^{n}\{\tau C_i, 2\tau C_i, 3\tau C_i, ...\}$$

be arranged in the increasing order as $S_1, S_2, S_3, ...$, such that $(0,\infty)$ is cut into subintervals $I_1 = (S_0, S_1]$, $I_2 = (S_1, S_2]$, $I_3 = (S_2, S_3]$, ..., where $S_0 = 0$. Notice that $S_j = \gamma_j \tau$ for all $j \ge 0$, where γ_j is determined by $C_1, C_2, ..., C_n$. We prove by induction on $j \ge 1$, that in the subinterval $I_j = (S_{j-1}, S_j]$, we have

$$T_{\rm TBS}(S) = \alpha_j(C_1, C_2, ..., C_n)S + \beta_j(C_1, C_2, ..., C_n)\tau,$$

where $\alpha_j(C_1, C_2, ..., C_n)$ is a positive function of $C_1, C_2, ..., C_n$ and $\beta_j(C_1, C_2, ..., C_n)$ is a nonnegative function of $C_1, C_2, ..., C_n$. First, it is clear that when $j = 1$ (i.e., for subinterval I_1), we have $S_1 = \tau C_1$ and

$$T_{\rm TBS}(S) = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right)S,$$

that is,

$$\alpha_1(C_1, C_2, ..., C_n) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i},$$

and

$$\beta_1(C_1, C_2, ..., C_n) = 0.$$

Next, we consider subinterval I_j with $j > 1$, based on the induction hypothesis that the expected file download time $T_{\rm TBS}(S)$ is

$$T_{\rm TBS}(S) = \alpha_{j'}(C_1, C_2, ..., C_n)S + \beta_{j'}(C_1, C_2, ..., C_n)\tau,$$

in subinterval $I_{j'}$, for all $1 \le j' \le j-1$. Let k_j be defined in such a way that $S > \tau C_i$ for all $1 \le i \le k_j$, and $S \le \tau C_i$

for all $k_j+1 \leq i \leq n$, where $1 \leq k_j \leq n$. The recurrence relation for $T_{\mathrm{TBS}}(S)$ in Theorem 1 can be rewritten as

$$T_{\mathrm{TBS}}(S) = \frac{1}{n}\left(\sum_{i=1}^{k_j}(\tau + T_{\mathrm{TBS}}(S-\tau C_i)) + \left(\sum_{i=k_j+1}^{n}\frac{1}{C_i}\right)S\right).$$

We notice that for all $S \in I_j$, the size $S-\tau C_i$ falls into I_{j_i} with $j_i < j$, where $1 \leq i \leq k_j$. By the induction hypothesis, we get

$$T_{\mathrm{TBS}}(S-\tau C_i) = \alpha_{j_i}(C_1, C_2, ..., C_n)(S-\tau C_i) + \beta_{j_i}(C_1, C_2, ..., C_n)\tau,$$

for all $1 \leq i \leq k_j$. Then, in subinterval I_j, we have

$$T_{\mathrm{TBS}}(S) = \frac{1}{n}\left(\sum_{i=1}^{k_j}(\tau + \alpha_{j_i}(C_1, C_2, ..., C_n)(S-\tau C_i) + \beta_{j_i}(C_1, C_2, ..., C_n)\tau) + \left(\sum_{i=k_j+1}^{n}\frac{1}{C_i}\right)S\right),$$

which is actually

$$\begin{aligned}
&T_{\mathrm{TBS}}(S) \\
&= \frac{1}{n}\left(\sum_{i=1}^{k_j}\alpha_{j_i}(C_1, C_2, ..., C_n) + \sum_{i=k_j+1}^{n}\frac{1}{C_i}\right)S \\
&\quad + \frac{1}{n}\left(\sum_{i=1}^{k_j}(1-\alpha_{j_i}(C_1, C_2, ..., C_n)C_i + \beta_{j_i}(C_1, C_2, ..., C_n))\right)\tau \\
&= \alpha_j(C_1, C_2, ..., C_n)S + \beta_j(C_1, C_2, ..., C_n)\tau,
\end{aligned}$$

where

$$\alpha_j(C_1, C_2, ..., C_n) = \frac{1}{n}\left(\sum_{i=1}^{k_j}\alpha_{j_i}(C_1, C_2, ..., C_n) + \sum_{i=k_j+1}^{n}\frac{1}{C_i}\right),$$

and

$$\beta_j(C_1, C_2, ..., C_n) = \frac{1}{n}\left(\sum_{i=1}^{k_j}(1-\alpha_{j_i}(C_1, C_2, ..., C_n)C_i + \beta_{j_i}(C_1, C_2, ..., C_n))\right).$$

This proves the theorem. ∎

III. Performance Comparison

Now, we are ready to compare the performance of algorithm TBS with algorithms PC and CBS. The following theorem states that the expected file download time of the time-based switching algorithm is no longer than that of the permanent connection algorithm and the chunk-based switching algorithm, where the equality holds only for homogeneous source peers or files of very small sizes.

Theorem 3: For any file S, we have $T_{\mathrm{TBS}}(S) \leq T_{\mathrm{PC}}(S) = T_{\mathrm{CBS}}(S)$, where the equality holds only for homogeneous source peers or $S \leq \tau \min\{C_1, C_2, ..., C_n\}$.

Proof. We prove the theorem by induction on S. First, it is easy to observe that

$$T_{\mathrm{TBS}}(S) = T_{\mathrm{PC}}(S) = T_{\mathrm{CBS}}(S),$$

for all $0 \leq S \leq \tau \min\{C_1, C_2, ..., C_n\}$. Next, for arbitrary $S > \tau \min\{C_1, C_2, ..., C_n\}$, we are going to show that $T_{\mathrm{TBS}}(S) \leq T_{\mathrm{PC}}(S) = T_{\mathrm{CBS}}(S)$ based on the induction hypothesis that $T_{\mathrm{TBS}}(S') \leq T_{\mathrm{PC}}(S') = T_{\mathrm{CBS}}(S')$ for all $S' < S$. Without loss of generality, we assume that $S > \tau C_i$ for all $1 \leq i \leq k$, and $S \leq \tau C_i$ for all $k+1 \leq i \leq n$, where $1 \leq k \leq n$. The recurrence relation for $T_{\mathrm{TBS}}(S)$ in Theorem 1 can be rewritten as

$$T_{\mathrm{TBS}}(S) = \frac{1}{n}\left(\sum_{i=1}^{k}(\tau + T_{\mathrm{TBS}}(S-\tau C_i)) + \sum_{i=k+1}^{n}\frac{S}{C_i}\right).$$

By the induction hypothesis, we have

$$\begin{aligned}
T_{\mathrm{TBS}}(S-\tau C_i) &\leq T_{\mathrm{PC}}(S-\tau C_i) \\
&= T_{\mathrm{CBS}}(S-\tau C_i) \\
&= (S-\tau C_i)\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right),
\end{aligned}$$

for all $1 \leq i \leq k$. Hence, we get

$$\begin{aligned}
T_{\mathrm{TBS}}(S) &\leq \frac{1}{n}\left(\sum_{i=1}^{k}\left(\tau + (S-\tau C_i)\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right)\right) + \sum_{i=k+1}^{n}\frac{S}{C_i}\right) \\
&= \frac{1}{n}\left(k\tau + \left(\sum_{i=1}^{k}(S-\tau C_i)\right)\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right)\right.
\end{aligned}$$

$$
\begin{aligned}
&\qquad + \sum_{i=k+1}^{n} \frac{S}{C_i}\Bigg) \\
&= \frac{1}{n}\left(k\tau + \left(kS - \tau\sum_{i=1}^{k} C_i\right)\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i}\right)\right. \\
&\qquad \left. + \sum_{i=k+1}^{n}\frac{S}{C_i}\right).
\end{aligned}
$$

Since

$$
\frac{1}{n}\sum_{i=1}^{n}\frac{1}{C_i} \le \frac{1}{k}\sum_{i=1}^{k}\frac{1}{C_i},
$$

where the equality holds only when $k = n$, we obtain

$$
\begin{aligned}
T_{\mathrm{TBS}}(S) &\le \frac{1}{n}\left(k\tau + \left(kS - \tau\sum_{i=1}^{k} C_i\right)\left(\frac{1}{k}\sum_{i=1}^{k}\frac{1}{C_i}\right)\right. \\
&\qquad \left. + \sum_{i=k+1}^{n}\frac{S}{C_i}\right) \\
&= \frac{1}{n}\left(k\tau - \left(\tau\sum_{i=1}^{k} C_i\right)\left(\frac{1}{k}\sum_{i=1}^{k}\frac{1}{C_i}\right)\right. \\
&\qquad \left. + \sum_{i=1}^{k}\frac{S}{C_i} + \sum_{i=k+1}^{n}\frac{S}{C_i}\right) \\
&= \frac{1}{n}\left(k\tau\left(1 - \left(\frac{1}{k}\sum_{i=1}^{k} C_i\right)\left(\frac{1}{k}\sum_{i=1}^{k}\frac{1}{C_i}\right)\right)\right. \\
&\qquad \left. + \sum_{i=1}^{n}\frac{S}{C_i}\right).
\end{aligned}
$$

Since

$$
\left(\frac{1}{k}\sum_{i=1}^{k} C_i\right) \ge \frac{1}{\left(\frac{1}{k}\sum_{i=1}^{k}\frac{1}{C_i}\right)},
$$

that is, the harmonic mean of the C_i's is no greater than the arithmetic mean of the C_i's, where the equality holds only when $C_1 = C_2 = \cdots = C_k$, we have

$$
\left(\frac{1}{k}\sum_{i=1}^{k} C_i\right)\left(\frac{1}{k}\sum_{i=1}^{k}\frac{1}{C_i}\right) \ge 1,
$$

which yields

$$
T_{\mathrm{TBS}}(S) \le \frac{1}{n}\sum_{i=1}^{n}\frac{S}{C_i} = T_{\mathrm{PC}}(S) = T_{\mathrm{CBS}}(S).
$$

Notice that $T_{\mathrm{TBS}}(S) = T_{\mathrm{PC}}(S) = T_{\mathrm{CBS}}(S)$ only when $C_1 = C_2 = \cdots = C_k$ with $k = n$, that is, the n source peers are homogeneous. The theorem is proven. ∎

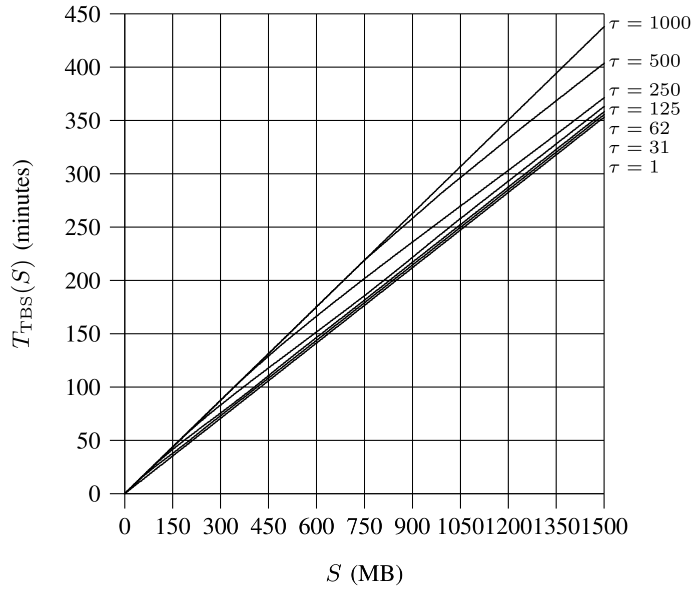

Figure 1. The expected file download time $T_{\mathrm{TBS}}(S)$ vs. file size S.

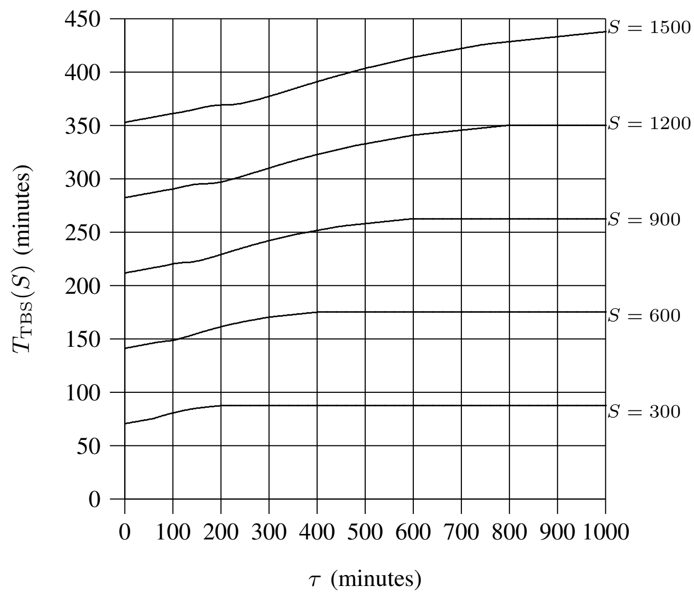

Figure 2. The expected file download time $T_{\mathrm{TBS}}(S)$ vs. period τ.

IV. Numerical Data

In this section, we present a numerical example to compare the performance of algorithms PC, CBS, and TBS. As in most P2P file sharing and exchange systems, the file sizes are in the range $10 \sim 1500$ MB [1]. The service capacity of a source peer is in the range $50 \sim 1,000$ kbps, i.e., $0.375 \sim 7.5$ MB/min.

Let us consider a P2P file sharing system with $n = 12$ source peers. The service capacities of the n source peers are 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0 MB/min.

In Figure 1, we demonstrate the expected file download time of the time-based switching algorithm as a function of file size S, where $0 \le S \le 1500$, for τ =1000, 500, 250, 125, 62, 31, 1.

In Figure 2, we demonstrate the expected file download time of the time-based switching algorithm as a function of the length of time period τ, where $0 \le \tau \le 1000$, for S =300, 600, 900, 1200, 1500.

It is observed from Figures 1 and 2 that shorter time period τ yields better performance, i.e., shorter expected file download time. When $\tau = 1000$, i.e.,

$$S \le \tau \min\{C_1, C_2, ..., C_n\},$$

the random time-based switching algorithm has the same expected file download time as the random permanent connection algorithm and the random chunk-based switching algorithm. When $\tau = 1$, i.e., almost zero, the random time-based switching algorithm approaches its minimum expected file download time.

The following facts formally describe our observations, which provide further details to Theorem 3.

Fact 1: For a fixed S and $n \ge 2$ heterogeneous source peers, if

$$\tau < \max\left\{\frac{S}{C_1}, \frac{S}{C_2}, ..., \frac{S}{C_n}\right\} = \frac{S}{\min\{C_1, C_2, ..., C_n\}},$$

the expected file download time $T_{\rm TBS}(S)$ of the random time-based switching algorithm is a nondecreasing function of τ.

Fact 2: If the period τ is

$$\tau \ge \max\left\{\frac{S}{C_1}, \frac{S}{C_2}, ..., \frac{S}{C_n}\right\} = \frac{S}{\min\{C_1, C_2, ..., C_n\}},$$

the expected file download time $T_{\rm TBS}(S)$ is

$$T_{\rm TBS}(S) = \frac{S}{\left(\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\frac{1}{C_i}\right)^{-1}},$$

namely, the file size divided by the harmonic mean of service capacities, which is the same as that of the random permanent connection algorithm and the random chunk-based switching algorithm.

Fact 3: As the period τ approaches zero, the expected file download time $T_{\rm TBS}(S)$ approaches its minimum value which is

$$\lim_{\tau\to 0} T_{\rm TBS}(S) = \frac{S}{\dfrac{1}{n}(C_1 + C_2 + \cdots + C_n)},$$

namely, the file size divided by the arithmetic mean of service capacities.

Fact 4: The ratios $T_{\rm TBS}(S)/T_{\rm PC}(S)$ and $T_{\rm TBS}(S)/T_{\rm CBS}(S)$ can be arbitrarily small, that is, the performance of the random permanent connection algorithm and the random chunk-based switching algorithm can be arbitrarily worse than the performance of the random time-based switching algorithm.

V. Conclusions

We have analyzed the expected file download time of the time-based switching algorithm for file sharing in P2P networks when the service capacity of a source peer is totally correlated over time, namely, the service capacities of a source peer in different time slots are a fixed value. We have developed a recurrence relation to characterize the expected file download time of the time-based switching algorithm. We have shown that the expected file download time of the time-based switching algorithm is in the range of the file size divided by the harmonic mean of service capacities and the file size divided by the arithmetic mean of service capacities, i.e.,

$$\frac{S}{\dfrac{1}{n}(C_1 + C_2 + \cdots + C_n)} < T_{\rm TBS}(S) \le \frac{S}{\left(\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\frac{1}{C_i}\right)^{-1}}.$$

Unless for very long time period, the time-based switching algorithm performs better than the permanent connection algorithm and the chunk-based switching algorithm.

It is an interesting and important open problem to analyze the expected file download time of the time-based switching algorithm for file sharing in P2P networks when the service capacities of the source peers are random variables.

REFERENCES

[1] http://blogplots.blogspot.com/2008/02/p2p-file-size-distribution.html

[2] http://en.wikipedia.org/wiki/File_sharing

[3] http://en.wikipedia.org/wiki/Peer-to-peer

[4] M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Suel, and D. D. Yao, "Optimal peer selection for P2P downloading and streaming," *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1538-1549, 2005.

[5] M. Adler, R. Kumar, K. Ross, D. Rubenstein, D. Turner, and D. D. Yao, "Optimal peer selection in a free-market peer-resource economy," *Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[6] D. S. Bernstein, Z. Feng, B. N. Levine, and S. Zilberstein, "Adaptive peer selection," *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems*, 2003.

[7] R. L. Carter and M. E. Crovella, "On the network impact of dynamic server selection," *Computer Networks*, vol. 31, no. 23-24, pp. 2529-2558, 1999.

[8] Y.-M. Chiu and D. Y. Eun, "Minimizing file download time in stochastic peer-to-peer networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 253-266, 2008.

[9] S. G. Dykes, K. A. Robbins, and C. L. Jeffery, "An empirical evaluation of client-side server selection algorithms," *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1361-1370, 2000.

[10] R. Gaeta, M. Gribaudo, D. Manini, and M. Sereno, "Analysis of resource transfers in peer-to-peer file sharing applications using fluid models," *Performance Evaluation*, vol. 63, pp. 149-174, 2006.

[11] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling peer-peer file sharing systems," *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 2188-2198, 2003.

[12] S. G. M. Koo, K. Kannan, C. S. G. Lee, "On neighbor-selection strategy in hybrid peer-to-peer networks," *Future Generation Computer Systems*, vol. 22, pp. 732-741, 2006.

[13] R. Kumar and K. W. Ross, "Peer-assisted file distribution: the minimum distribution time," *Proceedings of the IEEE Workshop on Hot Topics in Web Systems and Technologies*, 2006.

[14] M. Lingjun and K.-S. Lui, "Scheduling in P2P file distribution - On reducing the average distribution time," *Proceedings of the 5th IEEE Consumer Communications and Networking Conference*, pp. 521-522, 2008.

[15] M. Lingjun, W. Xiaolei, and K.-S. Lui, "A novel peer grouping scheme for P2P file distribution networks," *Proceedings of the IEEE International Conference on Communications*, pp. 5598-5602, 2008.

[16] M. Teo, C. Carbunaru, B. Leong, Y. Nataraj, H. M. L. Vu, R. Tan, Y. M. Teo, "Achieving high-bandwidth peer-to-peer file distribution," *Proceedings of the 4th ACM International Conference on Emerging Networking Experiments and Technologies*, 2008.

[17] S. Tewari and L. Kleinrock, "On fairness, optimal download performance and proportional replication in peer-to-peer networks," *Proceedings of the 4th International IFIP-TC6 Networking Conference* (LNCS 3462), pp. 709-717, 2005.

[18] X. Zheng, C. Cho, and Y. Xia, "Optimal peer-to-peer technique for massive content distribution," *Proceedings of the 27th IEEE Conference on Computer Communications*, pp. 151-155, 2008.