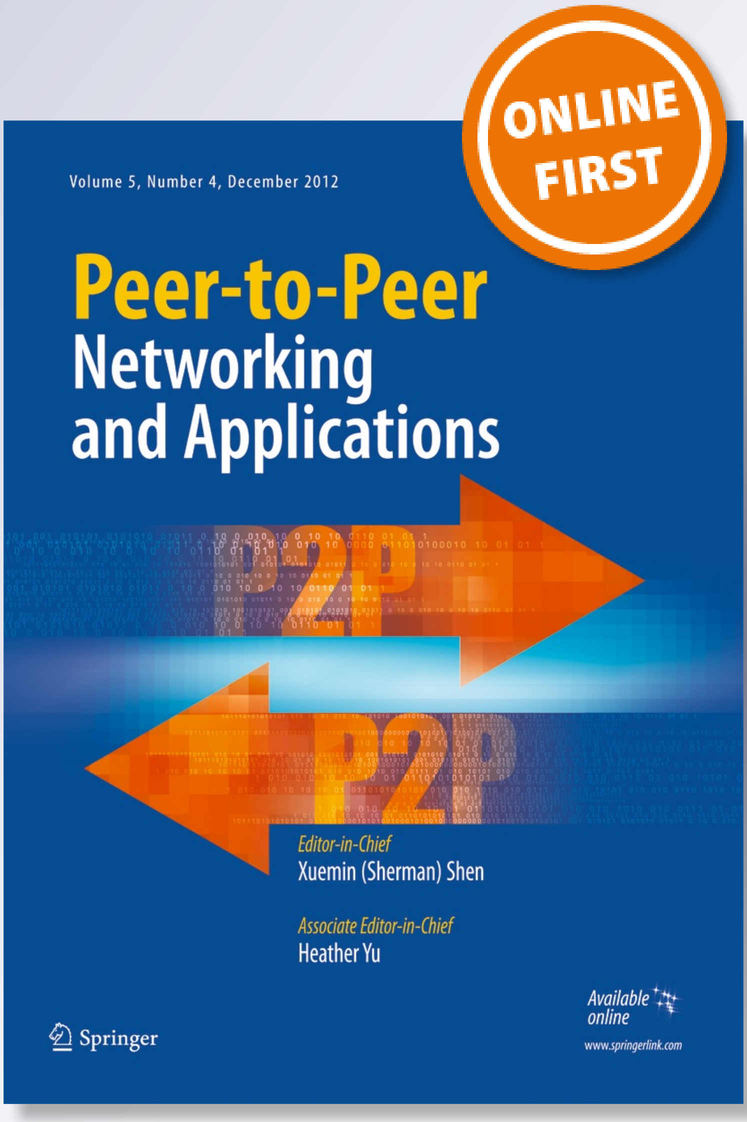
# *On the expected file download time of the random time-based switching algorithm in P2P networks*

**Keqin Li**

**Peer-to-Peer Networking and Applications**

ISSN 1936-6442

Peer-to-Peer Netw. Appl.
DOI 10.1007/s12083-012-0188-9



Peer-to-Peer Netw. Appl.
DOI 10.1007/s12083-012-0188-9

# On the expected file download time of the random time-based switching algorithm in P2P networks

**Keqin Li**

Received: 26 July 2011 / Accepted: 21 November 2012

**Abstract** The expected file download time of the random 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. It 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

K. Li (✉)
Department of Computer Science, State University of New York, New Paltz, NY 12561, USA
e-mail: lik@newpaltz.edu

## 1 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 P2P network provides services in a way different from that of centralized resources where a small number of servers provide all services. A pure P2P network does not have the notion of clients and servers, but only equal peers that simultaneously function as both clients and servers to other peers. This model of network architecture differs from the traditional client-server model where communication is among many clients and a single central server.

A unique feature of P2P networks is that all peers contribute resources, including storage space, computing power, and communication bandwidth. Therefore, as participating peers in a network increases, the total service capacity of the network also increases. This is not the case in a client-server system with a small number of servers, where adding more clients reduces the speed of data transfer for all clients and degrades the overall system performance. In addition to the above 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. In a pure P2P network, peers find locations of data without relying on a centralized index server, which means that there is no single point of failure in the network.

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. File sharing means distributing and accessing digitally stored information such as computer programs, multimedia, documents, and electronic books. It can be implemented in various storage, transmission, and distribution models. Common file sharing methods are manual sharing using removable memory, centralized file servers, WWW-based hyperlinked documents, and distributed P2P networking. 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 [12, 26] and fluid models [11], 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 user peers so that the overall distribution time is minimized [14, 16, 20, 21, 25, 27]. At the individual peer level, research is focused on analyzing and minimizing the file download time of a single peer [9, 17, 18].

It is clear that the vast majority of file downloads are performed by individual users. Fine system level modeling and efficient group file distribution methods do not help individual users to minimize their file download times. 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 [8, 10]. 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 [9], 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 [9], 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 arithmetic 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.

The motivation and significance of our study is twofold. First, there has been no analysis for the time-based switching algorithm when service capacities are highly or totally correlated. We contribute new findings in this direction. Second, such analysis is necessary when the time-based switching algorithm is to be compared with other algorithms, such as the permanent connection algorithm and the chunk-based switching algorithm. We reveal the fact that the time-based switching algorithm does perform better than the other two methods.

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.

Our analytical results are very clean and have intuitive explanations. For very long length of time period, the time-based switching algorithm behaves no differently from the permanent connection algorithm and the chunk-based switching algorithm, because all source peers are chosen with the same chance (or, equivalent, roughly the same percentage of a file is downloaded from all source peers). As the length of time period decreases, the source peers have increased chance of being used for the same amount of time. In the ideal case, when all source peers are chosen with the same amount of time $t$, the download time $T$ must be the file size divided by the arithmetic mean of service capacities, since the time $t$ is the download time $T$ divided by the number of source peers, and the file size is the product of $t$ and the total service capacity.

In addition to analytical results, we also present numerical examples and data to demonstrate our analytical results. Furthermore, we derive approximate closed form expressions for the expected file download time of the time-based switching algorithm and demonstrate that they are very accurate.

We notice that it is very effective to employ chunk-based switching and peer selection by using the method of probing high-capacity peers [18]. Furthermore, the method of parallel downloading has been used in reducing file download times [7, 9, 13, 15, 19, 22, 23]. However, these topics are beyond the scope of this paper. Due to space limitation, analysis of and comparison with these methods deserve separate papers. We will propose and analyze and compare the methods of probing high-capacity peers and parallel file download algorithms in P2P networks with random service capacities in separate papers. The focus of this paper is to analyze the random time-based switching algorithm and compare its performance with the random chunk-based switching algorithm.

## 2 Algorithms and analysis

Assume that $n$ peers $1, 2, \ldots, n$ 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 of source peer $i$ is $C_i$, which is unknown when a file is to be downloaded. The $n$ source peers are homogeneous if $C_1 = C_2 = \cdots = C_n$.

It is assumed that all source peers are stable and remain in a P2P network for significant amount of time. There is no effect of peer churn [24] for downloading the file of interest, i.e., all source peers are available during downloading of the file. Moreover, all the $n$ source peers are seed peers, i.e., they all hold a complete copy of a file, such that any chunk of the file can be obtained from any source peer. Further analysis of the effect of peer churn and non-seed peers can be a direction for future investigation.

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 $C_1, C_2, \ldots, C_n$.

### 2.1 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 $C_1, C_2, \ldots, C_n$.

**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

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

This proves the result. □

### 2.2 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 $1, 2, \ldots, m$. 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 $T_{\mathrm{CBS}}(S)$ of the random chunk-based switching algorithm is*

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

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

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

This proves the result. □

## 2.3 Random time-based switching

The random *time-based switching* (TBS) algorithm divides time into periods of equal length $\tau$. 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 $T_{\mathrm{TBS}}(S)$.

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

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

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

*Proof* Consider the first period $\tau$. 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 $\tau C_i$ is downloaded from source peer $i$, and the expected download time for the rest of the file of size $S-\tau C_i$ is $T_{\mathrm{TBS}}(S-\tau C_i)$, and the total download time is $\tau + T_{\mathrm{TBS}}(S-\tau C_i)$. 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 $T_{\mathrm{TBS}}(S)$ of the random time-based switching algorithm is

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

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 $T_{\mathrm{TBS}}(S)$. To show this, consider $n = 2$ source peers with $C_1 = C$ and $C_2 = 2C$. It is easy to verify that

$$\begin{aligned} T_{\mathrm{TBS}}(S) &= \frac{1}{2}\left(\frac{S}{C_1}+\frac{S}{C_2}\right) \\ &= \frac{1}{2}\left(\frac{S}{C}+\frac{S}{2C}\right) \\ &= 0.75\frac{S}{C}, \quad 0 < S \le \tau C; \\ T_{\mathrm{TBS}}(S) &= \frac{1}{2}\left(\tau + T_{\mathrm{TBS}}(S-\tau C_1) + \frac{S}{C_2}\right) \\ &= \frac{1}{2}\left(\tau + T_{\mathrm{TBS}}(S-\tau C) + \frac{S}{2C}\right) \\ &= \frac{1}{2}\left(\tau + 0.75\frac{S-\tau C}{C} + 0.5\frac{S}{C}\right) \\ &= \frac{1}{2}\left(1.25\frac{S}{C} + 0.25\tau\right) \\ &= 0.625\frac{S}{C} + 0.125\tau, \quad \tau C < S \le 2\tau C; \\ T_{\mathrm{TBS}}(S) &= \frac{1}{2}\left(\tau + T_{\mathrm{TBS}}(S-\tau C_1) + \tau + T_{\mathrm{TBS}}(S-\tau C_2)\right) \\ &= \frac{1}{2}\left(\tau + T_{\mathrm{TBS}}(S-\tau C) + \tau + T_{\mathrm{TBS}}(S-2\tau C)\right) \\ &= \frac{1}{2}\left(\tau + 0.625\frac{S-\tau C}{C} + 0.125\tau \right. \\ &\qquad \left. + \tau + 0.75\frac{S-2\tau C}{C}\right) \\ &= 0.6875\frac{S}{C}, \quad 2\tau C < S \le 3\tau C; \end{aligned}$$

$$\begin{aligned}
T_{\mathrm{TBS}}(S) &= \frac{1}{2}\left(\tau + T_{\mathrm{TBS}}(S-\tau C_1) + \tau + T_{\mathrm{TBS}}(S-\tau C_2)\right)\\
&= \frac{1}{2}\left(\tau + T_{\mathrm{TBS}}(S-\tau C) + \tau + T_{\mathrm{TBS}}(S-2\tau C)\right)\\
&= \frac{1}{2}\Bigg(\tau + 0.6875\frac{S-\tau C}{C}\\
&\qquad + \tau + 0.625\frac{S-2\tau C}{C} + 0.125\tau\Bigg)\\
&= \frac{1}{2}\left(1.3125\frac{S}{C} + 0.1875\tau\right)\\
&= 0.65625\frac{S}{C} + 0.09375\tau, \quad 3\tau C < S \le 4\tau C;\\
&\ \vdots
\end{aligned}$$

The above calculation shows that $T_{\mathrm{TBS}}(S)$ has a different expression in each subinterval $((j-1)\tau C, j\tau C]$, where $j \ge 1$. If we represent $T_{\mathrm{TBS}}(S)$ as $\alpha_j(S/C) + \beta_j\tau$ in subinterval $((j-1)\tau C, j\tau C]$, then as $j \to \infty$, we have $\alpha_\infty = 0.6666666....$

Although there is no closed form expression of $T_{\mathrm{TBS}}(S)$, we at least know that the expected file download time $T_{\mathrm{TBS}}(S)$ is a piecewise linear function of $S$. Let $(0,\infty)$ be cut into subintervals $I_1 = (S_0, S_1]$, $I_2 = (S_1, S_2]$, $I_3 = (S_2, S_3]$, ..., $I_j = (S_{j-1}, S_j]$, ..., where $S_0 < S_1 < S_2 < S_3 < \cdots < S_j < \cdots$, and $S_j = \gamma_j\tau$ for all $j \ge 0$, with $\gamma_j$ determined by $C_1, C_2, ..., C_n$. Without loss of generality, we assume that $C_1 \le C_2 \le \cdots \le C_n$. The values of $S_0, S_1, S_2, S_3, ..., S_j, ...$ are defined inductively as follows. First, we have $S_0 = 0$ and $S_1 = \tau C_1$. Based on $S_0, S_1, S_2, S_3, ..., S_{j-1}$, we define $S_j$ to be largest value which satisfies the following condition, namely, for all $S_{j-1} < S \le S_j$, the size $S - \tau C_i$ falls into $I_{j_i}$ with $j_i < j$, where $1 \le i \le k_j$, and $k_j$ is the largest integer such that for all $S_{j-1} < S \le S_j$, we have $S > \tau C_i$ for all $1 \le i \le k_j$, and $S \le \tau C_i$ for all $k_j + 1 \le i \le n$.

**Theorem 2** *The expected file download time $T_{\mathrm{TBS}}(S)$ is a piecewise linear function of $S$ in the subintervals of* $(0,\infty)$ *cut by* $S_0, S_1, S_2, S_3, ..., S_j, ....$

*Proof* We prove by induction on $j \ge 1$, that in the subinterval $I_j = (S_{j-1}, S_j]$, we have

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

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

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

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 $T_{\mathrm{TBS}}(S)$ is

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

in subinterval $I_{j'}$, for all $1 \le j' \le j-1$. Recall that $k_j$ is 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 $k_j + 1 \le i \le n$, where $1 \le k_j \le n$. 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 $1 \le i \le k_j$. By the induction hypothesis, we get

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

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

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

which is actually

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

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. □

## 3 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 (or equivalently, very long length of time period $\tau$).

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

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

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

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

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

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

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

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

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

**Table 1** Comparison of Analytical and Simulation Data of $T_{\mathrm{TBS}}(S)$ ($\tau = 15$)

| $S$ | Analytical data | Simulation data | Relative error (%) |
|---|---|---|---|
| 100 | 24.63 | 24.66 | 0.12 |
| 150 | 36.56 | 36.57 | 0.03 |
| 200 | 48.29 | 48.22 | −0.14 |
| 250 | 60.06 | 60.00 | −0.10 |
| 300 | 71.83 | 71.83 | 0.00 |
| 350 | 83.59 | 83.57 | −0.02 |
| 400 | 95.35 | 95.30 | −0.05 |
| 450 | 107.12 | 107.01 | −0.10 |
| 500 | 118.88 | 118.78 | −0.08 |
| 550 | 130.65 | 130.58 | −0.05 |
| 600 | 142.41 | 142.48 | 0.05 |
| 650 | 154.18 | 154.15 | −0.02 |
| 700 | 165.94 | 165.89 | −0.03 |
| 750 | 177.71 | 177.65 | −0.03 |
| 800 | 189.47 | 189.54 | 0.04 |
| 850 | 201.24 | 201.16 | −0.04 |
| 900 | 213.00 | 212.94 | −0.03 |
| 950 | 224.77 | 224.77 | 0.00 |
| 1000 | 236.53 | 236.43 | −0.04 |
| 1050 | 248.30 | 248.19 | −0.04 |
| 1100 | 260.06 | 259.93 | −0.05 |
| 1150 | 271.83 | 271.76 | −0.03 |
| 1200 | 283.59 | 283.67 | 0.03 |
| 1250 | 295.35 | 295.38 | 0.01 |
| 1300 | 307.12 | 307.19 | 0.02 |
| 1350 | 318.88 | 318.89 | 0.00 |
| 1400 | 330.65 | 330.55 | −0.03 |
| 1450 | 342.41 | 342.29 | −0.04 |
| 1500 | 354.18 | 354.00 | −0.05 |

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. □

## 4 Numerical and simulation 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 ∼ 1500 MB [1]. The service capacity of a source peer is in the range 50 ∼ 1,000 kbps, i.e., 0.375 ∼ 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 Table 1, we show simulation results and their accuracy compared with the analytical results for the expected file download time of the time-based switching algorithm. The time period is $\tau = 15$ min. The analytical data are calculated by using Theorem 1. Each simulation datum is obtained by repeating a simulation (i.e., execution of the time-based switching algorithm) for sufficient number of times, such that the 99 % confidence interval is sufficiently small, which is ±1.03598 % in our case. The relative error of each simulation datum is also given, which is no greater than ±0.14 %. These simulation data validate the correctness of our analytical data.

In Fig. 1, we demonstrate the expected file download time of the time-based switching algorithm as a function

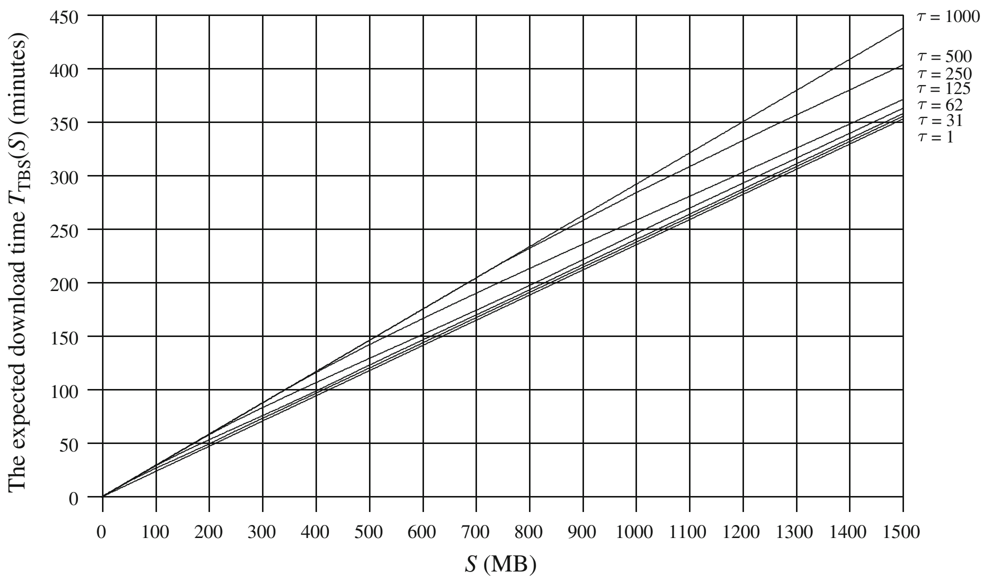


**Fig. 1** The expected file download time $T_{\mathrm{TBS}}(S)$ vs. file size $S$.

of file size $S$, where $0 \leq S \leq 1500$, for $\tau = 1000$, 500, 250, 125, 62, 31, 1 min. Notice that in Fig. 1, the curve for $\tau = 1000$ is precisely the performance of the permanent connection algorithm and the chunk-based switching algorithm.

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

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

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

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** *If the time period $\tau$ is*

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

*the expected file download time $T_{\mathrm{TBS}}(S)$ of the random time-based switching algorithm is*

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

*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.*

*Proof* Fact 1 is a direct consequence of Theorem 1, from which we know that if

$$\tau \geq \max\left\{\frac{S}{C_1}, \frac{S}{C_2}, ..., \frac{S}{C_n}\right\},$$

we have

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

since $S \leq \tau C_i$, for all $1 \leq i \leq n$. □

**Fact 2** *For a fixed $S$ and $n \geq 2$ 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 $T_{\mathrm{TBS}}(S)$ is an increasing function of $\tau$.*

*Proof* From Theorem 2, we know that the expected file download time $T_{\mathrm{TBS}}(S)$ is a piecewise linear function of $\tau$, i.e.,

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

in the subinterval $S/\gamma_j \leq \tau < S/\gamma_{j-1}$, and hence, is a nondecreasing function of $\tau$. Without loss of generality, we assume that $C_1 \leq C_2 \leq \cdots \leq C_n$, i.e., $S/C_1 \geq S/C_2 \geq \cdots \geq S/C_n$. For $n \geq 2$ heterogeneous source peers, if

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

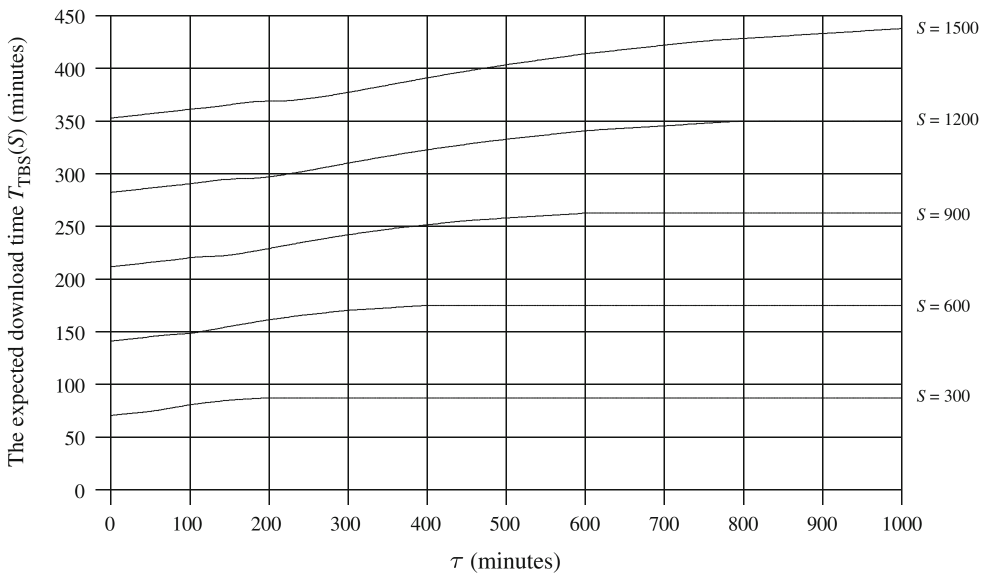


**Fig. 2** The expected file download time $T_{\mathrm{TBS}}(S)$ vs. period $\tau$.

there must be $k$, where $1 \leq k \leq n$, such that $S > \tau C_i$ for all $1 \leq i \leq k$, and $S \leq \tau C_i$ for all $k+1 \leq i \leq n$. By Theorem 1, we get

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

where, we can easily see that $T_{\text{TBS}}(S)$ is an increasing function of $\tau$, because all the $T_{\text{TBS}}(S - \tau C_i))$'s are at least nondecreasing functions of $\tau$. □

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

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

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

*Proof* As the length of the time period $\tau$ approaches zero, there will be increasing chance that all the source peers are used for the same amount of time. When all the source peers are used for the same amount of time $t$, we will have $T_{\text{TBS}}(S) = nt$, i.e., the expected file download time $T_{\text{TBS}}(S)$ is the number of source peers $n$ times the time $t$. Furthermore, we have $S = tC_1 + tC_2 + \cdots + tC_n = t(C_1 + C_2 + \cdots + C_n)$, which implies that

$$t = \frac{S}{C_1 + C_2 + \cdots + C_n},$$

i.e., the time $t$ is the file size divided by the total service capacity. Consequently, we get

$$T_{\text{TBS}}(S) = \frac{S}{\frac{1}{n}(C_1 + C_2 + \cdots + C_n)},$$

i.e., the expected file download time $T_{\text{TBS}}(S)$ is the file size divided by the arithmetic mean of service capacities. □

**Fact 4** *The ratios $T_{\text{TBS}}(S)/T_{\text{PC}}(S)$ and $T_{\text{TBS}}(S)/T_{\text{CBS}}(S)$ 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.*

*Proof* Fact 4 is based on the fact that the ratio of the arithmetic mean of service capacities to the harmonic mean of service capacities can be arbitrarily large. For instance, let us consider $n = 2$ source peers with $C_1 = 1$ and $C_2 = C$. The ratio of the arithmetic mean $(C+1)/2$ to the harmonic mean $2C/(C+1)$ is

$$\frac{(C+1)^2}{4C},$$

which can be arbitrarily large as $C \to 0$ or $C \to \infty$. Consequently, since by Fact 3,

$$T_{\text{TBS}}(S) \approx \frac{2S}{C+1} \text{ for very small } \tau,$$

and by Propositions 1 and 2,

$$T_{\text{PC}}(S) = T_{\text{CBS}}(S) = \frac{S(C+1)}{2C},$$

we get

$$T_{\text{TBS}}(S)/T_{\text{PC}}(S) = T_{\text{TBS}}(S)/T_{\text{CBS}}(S) \approx \frac{4C}{(C+1)^2},$$

which can be arbitrarily small when $C \to 0$ or $C \to \infty$. □

## 5 Closed form approximations

An approximate closed form expression for $T_{\text{TBS}}(S)$ can be obtained and its quality can be examined numerically. Assuming without loss of generality that $C_1 \leq C_2 \leq \cdots \leq C_n$. A good approximation of $T_{\text{TBS}}(S)$ is to consider the subinterval $(S_1, S_2]$, where $S_1 = \tau C_1$ and $S_2 = \min\{2\tau C_1, \tau C_2\}$, in which, we have

$$\begin{aligned}
T_{\text{TBS}}(S) &= \frac{1}{n}\left(\tau + T_{\text{TBS}}(S - \tau C_1) + \frac{S}{C_2} + \cdots + \frac{S}{C_n}\right)\\
&= \frac{1}{n}\left(\tau + (S - \tau C_1)\frac{1}{n}\left(\frac{1}{C_1} + \frac{1}{C_2} + \cdots + \frac{1}{C_n}\right)\right.\\
&\qquad \left. + S\left(\frac{1}{C_2} + \cdots + \frac{1}{C_n}\right)\right)\\
&= \frac{1}{n}\left(\left(\frac{1}{n}\left(\frac{1}{C_1} + \frac{1}{C_2} + \cdots + \frac{1}{C_n}\right)\right.\right.\\
&\qquad \left. + \left(\frac{1}{C_2} + \cdots + \frac{1}{C_n}\right)\right)S\\
&\qquad \left. + \left(1 - \frac{C_1}{n}\left(\frac{1}{C_1} + \frac{1}{C_2} + \cdots + \frac{1}{C_n}\right)\right)\tau\right).
\end{aligned}$$

For our example in the last section, the above approximation gives an over-estimation of $T_{\mathrm{TBS}}(S)$ by no more than 10.2 % for $\tau = 50$ and $100 \le S \le 1500$. It is very accurate for $S$ close to $\tau C_1$ or for $\tau$ close to $S/C_1$.

A more accurate approximation of $T_{\mathrm{TBS}}(S)$ can be obtained by considering the subinterval $(S_2, S_3]$, where $S_2$ and $S_3$ depend on the relationship among $C_1$, $C_2$, and $C_3$. For our example in the last section, we have $S_2 = C_2$ and $S_3 = C_3$. Therefore, we get

$$\begin{aligned}
T_{\mathrm{TBS}}(S) &= \frac{1}{n}\left(\tau + T_{\mathrm{TBS}}(S-\tau C_1) + \tau + T_{\mathrm{TBS}}(S-\tau C_2)\right.\\
&\qquad \left. + \frac{S}{C_3} + \cdots + \frac{S}{C_n}\right)\\
&= \frac{1}{n}\left(\tau + (S-\tau C_1)\frac{1}{n}\left(\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_n}\right)\right.\\
&\qquad + \tau + (S-\tau C_2)\frac{1}{n}\left(\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_n}\right)\\
&\qquad \left. + S\left(\frac{1}{C_3}+\cdots+\frac{1}{C_n}\right)\right)\\
&= \frac{1}{n}\left(\left(\frac{2}{n}\left(\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_n}\right)\right.\right.\\
&\qquad \left. + \left(\frac{1}{C_3}+\cdots+\frac{1}{C_n}\right)\right) S\\
&\qquad \left. + \left(2 - \frac{C_1+C_2}{n}\left(\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_n}\right)\right)\tau\right).
\end{aligned}$$

The above approximation gives an over-estimation of $T_{\mathrm{TBS}}(S)$ by no more than 3.8 % for $\tau = 50$ and $100 \le S \le 1500$.

If we further consider the subinterval $(S_3, S_4]$, where $S_3 = C_3$ and $S_4 = C_4$, we have

$$\begin{aligned}
T_{\mathrm{TBS}}(S) &= \frac{1}{n}\left(\left(\frac{3}{n}\left(\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_n}\right)\right.\right.\\
&\qquad \left. + \left(\frac{1}{C_4}+\cdots+\frac{1}{C_n}\right)\right) S\\
&\qquad + \left(3 - \frac{C_1+C_2+C_3}{n}\right.\\
&\qquad \left.\left. \times \left(\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_n}\right)\right)\tau\right).
\end{aligned}$$

The relative error of the above approximation is in the range [−0.09 %, 1.69 %] for $\tau = 50$ and $100 \le S \le 1500$. In Table 2, we give the numerical values of the above approximation and the actual values of the expected file download time $T_{\mathrm{TBS}}(S)$ when $\tau = 50$ for the same P2P file sharing system of Section 4.

**Table 2** Accurate approximation of $T_{\mathrm{TBS}}(S)$ ($\tau = 50$)

| $S$ | Actual value | Approximate value | Relative error (%) |
|---|---|---|---|
| 100 | 28.41 | 28.64 | 0.79 |
| 150 | 40.35 | 40.35 | 0.00 |
| 200 | 51.68 | 52.07 | 0.76 |
| 250 | 62.87 | 63.79 | 1.47 |
| 300 | 74.26 | 75.51 | 1.69 |
| 350 | 86.12 | 87.23 | 1.28 |
| 400 | 98.38 | 98.94 | 0.58 |
| 450 | 110.21 | 110.66 | 0.41 |
| 500 | 121.86 | 122.38 | 0.43 |
| 550 | 133.51 | 134.10 | 0.44 |
| 600 | 145.22 | 145.81 | 0.41 |
| 650 | 157.01 | 157.53 | 0.33 |
| 700 | 168.84 | 169.25 | 0.24 |
| 750 | 180.63 | 180.97 | 0.18 |
| 800 | 192.37 | 192.68 | 0.16 |
| 850 | 204.12 | 204.40 | 0.14 |
| 900 | 215.87 | 216.12 | 0.11 |
| 950 | 227.64 | 227.84 | 0.08 |
| 1000 | 239.42 | 239.56 | 0.06 |
| 1050 | 251.19 | 251.27 | 0.03 |
| 1100 | 262.95 | 262.99 | 0.02 |
| 1150 | 274.71 | 274.71 | −0.00 |
| 1200 | 286.47 | 286.43 | −0.02 |
| 1250 | 298.24 | 298.14 | −0.03 |
| 1300 | 310.01 | 309.86 | −0.05 |
| 1350 | 321.77 | 321.58 | −0.06 |
| 1400 | 333.54 | 333.30 | −0.07 |
| 1450 | 345.30 | 345.01 | −0.08 |
| 1500 | 357.06 | 356.73 | −0.09 |

## 6 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.,

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

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

The performance of the time-based switching algorithm is now well understood when the service capacities of a source peer in different time periods are lightly correlated or almost independent [9], or when the service capacities of a source peer in different time periods are highly or totally correlated, i.e., the service capacity of a source peer remains stable at least for a reasonable amount of time (e.g., within the time of downloading a file). However, the performance of the time-based switching algorithm is still not clear when the service capacities of a source peer in different time periods are partially correlated. It is therefore an interesting and important and challenging 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 a source peer in different time periods are partially correlated, i.e., to include temporal fluctuation of source peer capacities into consideration. New mathematical models are required to describe temporal correlation in the service capacity of a source peer. Such analysis is even more difficult if the effects of peer churn and non-seed peers are also included. It is conceivable that such investigation needs significantly new insights which are well beyond the scope of this paper, and deserves separate papers. In this sense, our effort in this paper is only an initial attempt towards this direction and should inspire subsequent studies.

**Acknowledgments** The criticism and comments from three reviewers are greatly appreciated.

## References

1. http://blogplots.blogspot.com/2008/02/p2p-file-size-distribution.html. Accessed 17 Dec 2012
2. http://en.wikipedia.org/wiki/File_sharing. Accessed 17 Dec 2012
3. http://en.wikipedia.org/wiki/Peer-to-peer. Accessed 17 Dec 2012
4. Adler M, Kumar R, Ross K, Rubenstein D, Suel T, Yao DD (2005) Optimal peer selection for P2P downloading and streaming. In: Proceedings of the 24th annual joint conference of the IEEE computer and communications societies, vol 3, pp 1538–1549
5. Adler M, Kumar R, Ross K, Rubenstein D, Turner D, Yao DD (2004) Optimal peer selection in a free-market peer-resource economy. In: Proceedings of the 2nd workshop on the economics of peer-to-peer systems
6. Bernstein DS, Feng Z, Levine BN, Zilberstein S (2003) Adaptive peer selection. In: Proceedings of the 2nd international workshop on peer-to-peer systems
7. Byers JW, Luby M, Mitzenmacher M (1999) Accessing multiple mirror sites in parallel: using Tornado codes to speed up downloads. In: Proceedings of the 18th annual joint conference of the IEEE computer and communications societies, vol 1, pp 275–283
8. Carter RL, Crovella ME (1999) On the network impact of dynamic server selection. Comput Networks 31(23–24):2529–2558
9. Chiu Y-M, Eun DY (2008) Minimizing file download time in stochastic peer-to-peer networks. IEEE/ACM Trans Netw 16(2):253–266
10. Dykes SG, Robbins KA, Jeffery CL (2000) An empirical evaluation of client-side server selection algorithms. In: Proceedings of the 19th annual joint conference of the IEEE computer and communications societies, vol 3, pp 1361–1370
11. Gaeta R, Gribaudo M, Manini D, Sereno M (2006) Analysis of resource transfers in peer-to-peer file sharing applications using fluid models. Perform Eval 63:149–174
12. Ge Z, Figueiredo DR, Jaiswal S, Kurose J, Towsley D (2003) Modeling peer-peer file sharing systems. In: Proceedings of the 22nd annual joint conference of the IEEE computer and communications societies, vol 3, pp 2188–2198
13. Gkantsidis C, Ammar M, Zegura E (2003) On the effect of large-scale deployment of parallel downloading. In: Proceedings of the 3rd IEEE workshop on internet applications, pp 79–89
14. Koo SGM, Kannan K, Lee CSG (2006) On neighbor-selection strategy in hybrid peer-to-peer networks. Future Gener Comput Syst 22:732–741
15. Koo SGM, Rosenberg C, Xu D (2003) Analysis of parallel downloading for large file distribution. In: Proceedings of the 9th IEEE workshop on future trends of distributed computing systems, pp 128–135
16. Kumar R, Ross KW (2006) Peer-assisted file distribution: the minimum distribution time. In: Proceedings of the IEEE workshop on hot topics in web systems and technologies
17. Li K (2010) Analysis of random time-based switching for file sharing in peer-to-peer networks. In: Proceedings of the 24th IEEE international parallel and distributed processing symposium workshops (7th international workshop on hot topics in peer-to-peer systems)
18. Li K (2011) Reducing download times in peer-to-peer file sharing systems with stochastic service capacities. In: Proceedings of the 25th IEEE international parallel and distributed processing symposium workshops (13th workshop on advances in parallel and distributed computational models), pp 603–612
19. Liu Y, Gong W, Shenoy P (2001) On the impact of concurrent downloads. In: Proceedings of the 33nd winter simulation conference, pp 1300–1305
20. Lingjun M, Lui K-S (2008) Scheduling in P2P file distribution—on reducing the average distribution time. In: Proceedings of the 5th IEEE consumer communications and networking conference, pp 521–522
21. Lingjun M, Xiaolei W, Lui K-S (2008) A novel peer grouping scheme for P2P file distribution networks. In: Proceedings of the IEEE international conference on communications, pp 5598–5602
22. Manini D, Gribaudo M (2006) Modelling search, availability, and parallel download in P2P file sharing applications with fluid model. In: Proceedings of 14th international conference on advanced computing and communications, pp 449–454
23. Rodriguez P, Biersack EW (2002) Dynamic parallel access to replicated content in the internet. IEEE/ACM Trans Netw 10(4):455–465
24. Stutzbach D, Rejaie R (2006) Understanding churn in peer-to-peer networks. In: Proceedings of the 6th ACM SIGCOMM conference on internet measurement
25. Teo M, Carbunaru C, Leong B, Nataraj Y, Vu HML, Tan R, Teo YM (2008) Achieving high-bandwidth peer-to-peer file distribution. In: Proceedings of the 4th ACM international conference on emerging networking experiments and technologies
26. Tewari S, Kleinrock L (2005) On fairness, optimal download performance and proportional replication in peer-to-peer networks. In: Proceedings of the 4th international IFIP-TC6 networking conference (LNCS), vol 3462, pp 709–717
27. Zheng X, Cho C, Xia Y (2008) Optimal peer-to-peer technique for massive content distribution. in: Proceedings of the 27th IEEE conference on computer communications, pp 151–155

**Keqin Li** is a SUNY Distinguished Professor. Professor Li's research interests are mainly in the areas of design and analysis of algorithms, parallel and distributed computing, and computer networking. He has contributed extensively to processor allocation and resource management; design and analysis of sequential/parallel, deterministic/probabilistic, and approximation algorithms; parallel and distributed computing systems performance analysis, prediction, and evaluation; job scheduling, task dispatching, and load balancing in heterogeneous distributed systems; dynamic tree embedding and randomized load distribution in static networks; parallel computing using optical interconnections; dynamic location management in wireless communication networks; routing and wavelength assignment in WDM optical networks; and energy-efficient power management and performance optimization. His current research interests include lifetime maximization in sensor networks, file sharing in peer-to-peer systems, and cloud computing. He has published over 245 journal articles, book chapters, and research papers in refereed international conference proceedings. He has received several Best Paper Awards for his highest quality work. He is currently on the editorial board of *IEEE Transactions on Parallel and Distributed Systems*, *Journal of Parallel and Distributed Computing*, *International Journal of Parallel, Emergent and Distributed Systems*, *International Journal of High Performance Computing and Networking*, and *Optimization Letters*.