International Journal of Foundations of Computer Science
Vol. 23, No. 6 (2012) 1341–1369

DOI: 10.1142/S0129054112500189



# PROBING HIGH-CAPACITY PEERS TO REDUCE DOWNLOAD TIMES IN P2P FILE SHARING SYSTEMS WITH STOCHASTIC SERVICE CAPACITIES

KEQIN LI

*Department of Computer Science, State University of New York*
*New Paltz, New York 12561, USA*
*lik@newpaltz.edu*

Received 22 July 2011
Accepted 28 February 2012
Communicated by Koji Nakano

The main problem for an individual user peer in a peer-to-peer network with heterogeneous source peers 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. There has been little investigation on selecting source peers with stochastic service capacities. The main contribution of this paper is to address the problem of reducing download times in peer-to-peer file sharing systems with stochastic service capacities. A precise analysis of the expected download time is given when the service capacity of a source peer is a random variable. A chunk-based switching and peer selection algorithm using the method of probing high-capacity peers is proposed and the expected download time of the algorithm is analyzed. Two subproblems of the optimal choice of the threshold of high-capacity source peers and the optimal order of probing are also solved. The performance of the algorithm is compared with the random chunk-based switching method. It is shown that noticeable performance improvement can be obtained.

*Keywords*: Download time; file sharing system; peer-to-peer network; peer selection; stochastic service capacity.

## 1. Introduction

A peer-to-peer (P2P) network employs diverse connectivity among participating peers and the combined resources of participants to provide various services [4]. 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 [13, 26] and fluid models [12], 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 [15, 17, 21, 22, 25, 27]. At the individual peer level, research is focused on analyzing and minimizing the file download time of a single peer [10, 18, 19].

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 this paper, we only consider reduction of 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 [7]. The problem is called the server selection problem in WWW client-server applications [9, 11]. The peer selection problem is also studied in the context of free-market economy with additional consideration of cost of download [5, 6].

Virtually all existing studies assume that the communication capacity between a pair of peers is a constant. Thus, the transfer time of a file of size $S$ from a source peer with service capacity $C$ is simply $S/C$. The assumption of constant service capacity is certainly not realistic, since a source peer has variable workload and a file transfer may encounter unpredictable network traffic and congestion and delay. Therefore, the performance of a peer selection policy based on such an assumption becomes vulnerable and unreliable. When the service capacity of a source peer is a random variable, the expected download time is not simply the file size divided by the expected service capacity. Unfortunately, there has been little investigation on selecting source peers with stochastic service capacities. In [10], the problem of minimizing file download time from source peers with time-varying service capacities is considered. A random chunk-based switching method is proposed, aiming to hide the heterogeneity of source peers and to achieve the harmonic mean of service capacities.

The main contribution of this paper is to address the problem of reducing download times in peer-to-peer file sharing systems with stochastic service capacities. We give a precise analysis of the expected download time when the service capacity of a source peer is a random variable (Section 2). We propose a chunk-based switching and peer selection algorithm using the method of probing high-capacity peers, and analyze the expected download time of our algorithm (Section 3). We also solve the two subproblems of the optimal choice of the threshold of high-capacity source peers (Section 5) and the optimal order of probing (Section 6). We compare the performance of our algorithm with the random chunk-based switching method of [10] and get noticeable performance improvement (Sections 4 and 6).

We notice that the method of parallel downloading has been used in reducing file download times [8, 10, 14, 16, 20, 23, 24]. However, this is beyond the scope of this paper and we will propose and analyze and compare parallel file download algorithms in P2P networks with random service capacities in a separate paper.

## 2. File Download Time

Throughout the paper, we use $\boldsymbol{P}[e]$ to denote the probability of an event $e$, $f_X(x)$ the probability distribution function (pdf), $F_X(x)$ the cumulative distribution function (cdf), and $\boldsymbol{E}(X)$ the expectation, respectively, of a random variable $X$.

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 further assume that the service capacity of source peer $i$ is $C_i$, a random variable in $[0, \infty)$ with pdf $f_{C_i}(c)$ and cdf $F_{C_i}(c)$. 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

$$T_i(S) = \frac{S}{C_i}.$$

Thus, we get the cdf of $T_i(S)$,

$$\begin{aligned} F_{T_i(S)}(t) &= \boldsymbol{P}[T_i(S) \le t] \\ &= \boldsymbol{P}\left[\frac{S}{C_i} \le t\right] \\ &= \boldsymbol{P}\left[C_i \ge \frac{S}{t}\right] \\ &= 1 - F_{C_i}\left(\frac{S}{t}\right), \end{aligned}$$

and the pdf of $T_i(S)$,

$$f_{T_i(S)}(t) = \frac{S}{t^2} f_{C_i}\left(\frac{S}{t}\right),$$

for all $t > 0$. Consequently, the expectation of $T_i(S)$ is

$$\begin{aligned} \boldsymbol{E}(T_i(S)) &= \int_0^\infty t f_{T_i(S)}(t))dt \\ &= \int_0^\infty \frac{S}{t} f_{C_i}\left(\frac{S}{t}\right) dt \\ &= \int_\infty^0 c f_{C_i}(c)\left(-\frac{S}{c^2}\right) dc \quad \left(\text{by letting } c = \frac{S}{t}\right) \\ &= \int_0^\infty \frac{S}{c} f_{C_i}(c)dc. \end{aligned}$$

Let $T(S, c) = S/c$ be the download time of a file of size $S$ from any source peer $i$ when $C_i = c$. The last equation can also be obtained by randomizing $c$ in $T_i(S, c)$ directly,

$$\boldsymbol{E}(T_i(S)) = \int_0^\infty T(S, c) f_{C_i}(c)dc = \int_0^\infty \frac{S}{c} f_{C_i}(c)dc.$$

The above equation can also be written as

$$\boldsymbol{E}(T_i(S)) = S \int_0^\infty \frac{f_{C_i}(c)}{c} dc = S\boldsymbol{E}(T_i(1)), \tag{1}$$

that is, $\boldsymbol{E}(T_i(S))$ is a linear function of $S$, where

$$\boldsymbol{E}(T_i(1)) = \int_0^\infty \frac{f_{C_i}(c)}{c} dc$$

is the expected download time of one unit of data from source peer $i$. Define a function $g(x) = 1/x$, which is a convex function. By the well known Jensen's inequality [3], we have

$$\boldsymbol{E}(T_i(1)) = \boldsymbol{E}(g(C_i)) \ge g(\boldsymbol{E}(C_i)) = \frac{1}{\boldsymbol{E}(C_i)},$$

for any $f_{C_i}(c)$, and

$$\boldsymbol{E}(T_i(S)) \geq \frac{S}{\boldsymbol{E}(C_i)},$$

for any $S$. The above inequality means that we cannot achieve the average service capacity of a source peer if we download a file at random time, which is a surprising claim. For instance, if $C_i$ has a uniform distribution in $[c_1, c_2]$, by straightforward calculation, we obtain $\boldsymbol{E}(T_i(1)) = \ln(c_2/c_1)/(c_2 - c_1)$ and $\boldsymbol{E}(C_i) = (c_1 + c_2)/2$. Thus,

$$\frac{\ln(c_2/c_1)}{c_2 - c_1} \geq \frac{2}{c_1 + c_2},$$

an inequality not obvious at all. (*Proof*: Let $x = c_2/c_1 \geq 1$. The above inequality becomes $\ln x/(x-1) \geq 2/(x+1)$, or, $\ln x \geq 2(x-1)/(x+1) = 2(1 - 2/(x+1))$, that is, $\ln x + 4/(x+1) \geq 2$. One can now show that the left hand side of the last inequality achieves its minimum value 2 when $x = 1$.) Furthermore, for a fixed mean $\boldsymbol{E}(C_i)$, $\boldsymbol{E}(T_i(1))$ increases as the variance of the uniform distribution increases, another claim which is not very obvious. (*Proof*: Let $c_1 = \mu - \sqrt{3}\sigma$ and $c_2 = \mu + \sqrt{3}\sigma$, where $\mu = \boldsymbol{E}(C_i) = (c_1 + c_2)/2$ and $\sigma^2$ is the variance. By defining $x = (\mu + \sqrt{3}\sigma)/(\mu - \sqrt{3}\sigma)$, we get $\boldsymbol{E}(T_i(1)) = y/(2\mu)$, where $y = ((x+1)/(x-1)) \ln x$. One can now show that $y$ is an increasing function of $x > 1$, where $x$ is an increasing function of $\sigma > 0$. )

In Figure 1, we demonstrate the relative difference between $\boldsymbol{E}(T_i(1))$ and $1/\boldsymbol{E}(C_i)$, that is,

$$\begin{aligned}\left(\frac{(\boldsymbol{E}(T_i(1)) - 1/\boldsymbol{E}(C_i))}{1/\boldsymbol{E}(C_i))}\right) \times 100\% &= (\boldsymbol{E}(T_i(1))\boldsymbol{E}(C_i) - 1) \times 100\% \\ &= \left(\frac{\mu}{2\sqrt{3}\sigma} \ln\left(\frac{\mu + \sqrt{3}\sigma}{\mu - \sqrt{3}\sigma}\right) - 1\right) \times 100\%\end{aligned}$$

for a uniform distribution of $C_i$ with mean $\boldsymbol{E}(C_i) = \mu = 4, 5, 6, 7$ and variance $\sigma^2$ with $\sigma$ in the range $(0, 2]$. It is observed that for a fixed mean $\mu$, the relative difference between $\boldsymbol{E}(T_i(1))$ and $1/\boldsymbol{E}(C_i)$ is an increasing function of variance $\sigma^2$ and increases more than linearly when $\sigma$ increases. Furthermore, for a fixed variance $\sigma^2$, the relative difference between $\boldsymbol{E}(T_i(1))$ and $1/\boldsymbol{E}(C_i)$ is a decreasing function of mean $\mu$. Therefore, the relative difference between $\boldsymbol{E}(T_i(1))$ and $1/\boldsymbol{E}(C_i)$ is an increasing function of the coefficient of variation $\sigma/\mu$.

We say that the $n$ source peers are homogeneous if their service capacities $C_1$, $C_2$, ..., $C_n$ are independent and identical random variables $C$ with the same pdf,

$$f_{C_1}(c) = f_{C_2}(c) = \cdots = f_{C_n}(c) = f_C(c),$$

and the same cdf,

$$F_{C_1}(c) = F_{C_2}(c) = \cdots = F_{C_n}(c) = F_C(c).$$

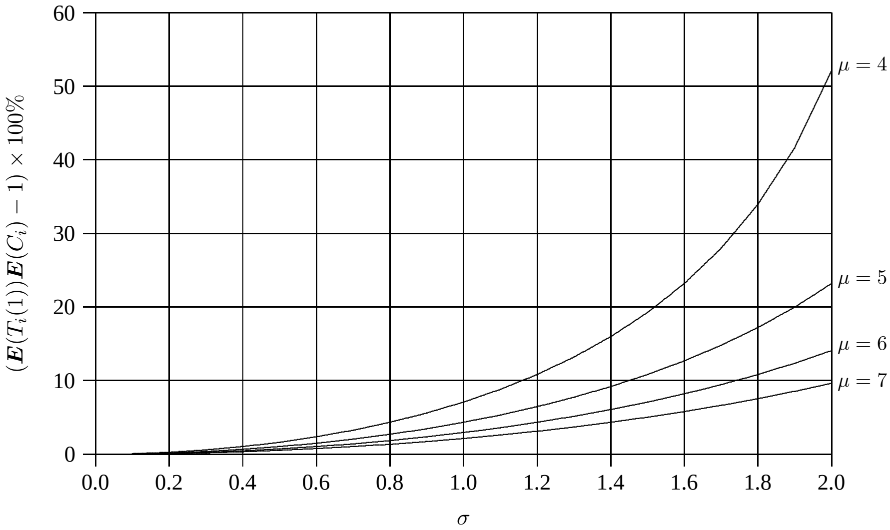


Fig. 1. The relative difference versus variance.

Notice that this does not mean that the $n$ source peers have the same service capacity. In fact, during transferring the same file at the same time, the service capacities of the $n$ source peers can be entirely and radically different as governed by $f_C(c)$.

For homogeneous source peers, we use $\boldsymbol{E}(T(S)) = S\boldsymbol{E}(T(1))$ to represent the expected download time of a file of size $S$ from any source peer, where

$$\boldsymbol{E}(T(1)) = \int_0^\infty \frac{f_C(c)}{c} dc$$

is the expected download time of one unit of data from any source peer.

## 3. Chunk-Based Switching and Peer Selection

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 source 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.

### 3.1. *Algorithm* $\mathbb{RP}$ *and analysis*

#### 3.1.1. *Algorithm*

In the random chunk-based switching algorithm $\mathbb{RP}$ (meaning: selecting a *R*andom *P*eer), a source peer $i_j \in \{1, 2, ..., n\}$ is randomly and uniformly chosen from

$\{1, 2, ..., n\}$ for each chunk $j$, where $1 \leq j \leq m$ [10]. Algorithm $\mathbb{RP}$ has no knowledge of and does not probe the current service capacities of the source peers.

3.1.2. *Analysis*

Let $T_{\mathbb{RP}}(S)$ denote the download time of algorithm $\mathbb{RP}$ for a file of size $S$. Then, we have

$$T_{\mathbb{RP}}(S) = \sum_{j=1}^{m} T_{i_j}(S^*).$$

The expected download time $\boldsymbol{E}(T_{i_j}(S^*))$ of a chunk $j$ from source peer $i_j = i$ is $\boldsymbol{E}(T_i(S^*))$. Since $i_j = i$ with probability $1/n$ for all $1 \leq i \leq n$, by Eq. (1), we have

$$\boldsymbol{E}(T_{i_j}(S^*)) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{E}(T_i(S^*)) = \frac{1}{n}\sum_{i=1}^{n} S^* \boldsymbol{E}(T_i(1)) = \frac{S^*}{n}\sum_{i=1}^{n} \boldsymbol{E}(T_i(1)),$$

for all $1 \leq j \leq m$. Since there are $m$ chunks, we obtain

$$\boldsymbol{E}(T_{\mathbb{RP}}(S)) = \sum_{j=1}^{m} \boldsymbol{E}(T_{i_j}(S^*)) = \frac{mS^*}{n}\sum_{i=1}^{n} \boldsymbol{E}(T_i(1)) = \frac{S}{n}\sum_{i=1}^{n} \boldsymbol{E}(T_i(1)),$$

where

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{E}(T_i(1))$$

is the expected download time of one unit of data when the $n$ source peers are chosen with equal probability. For homogeneous source peers, we have

$$\boldsymbol{E}(T_{\mathbb{RP}}(S)) = S\boldsymbol{E}(T(1)).$$

Assume that the file size $S$ is a random variable with pdf $f_S(s)$ in $[0, \infty)$. Let $T_{\mathbb{RP}}$ denote the download time of algorithm $\mathbb{RP}$ for a random file. Then, we have

$$\boldsymbol{E}(T_{\mathbb{RP}}) = \int_0^\infty \boldsymbol{E}(T_{\mathbb{RP}}(s)) f_S(s) ds.$$

### 3.2. *Algorithm* $\mathbb{HP}$ *and analysis*

3.2.1. *Algorithm*

Our algorithm $\mathbb{HP}$ (meaning: selecting a *H*igh-capacity *P*eer) for chunk-based switching and peer selection is given in Figure 2. A source peer $i$ is called a high-capacity source peer if $C_i \geq c^*$, and a source peer $i$ is called a low-capacity source peer if $C_i < c^*$, where $c^*$ is an appropriately chosen service capacity threshold. The choice of $c^*$ has strong impact on the performance of algorithm $\mathbb{HP}$, and we will address this important issue in a later section. Our algorithm consists of two stages. In the first stage (lines (1)–(9)), the source peers are probed one at a time by downloading one chunk from each source (lines (3) and (7)). This probing procedure is terminated under one of the following three conditions:

(1) All the chunks have been downloaded (i.e., $i = m + 1$) when $m \le n$ (line (2)), regardless whether a high-capacity source peer is found;
(2) A high-capacity source peer $i$ is identified (lines (4)–(5));
(3) All the $n$ source peers have been probed (i.e., $i = n + 1$) when $m > n$ and no high-capacity source peer is found (line (2)).

The second stage of the algorithm (lines (10)–(20)) completes the downloading of the remaining chunks. In the first case, there is no further work to be performed. In the second case, all the remaining chunks are downloaded from the high-capacity source peer $i$ found in the first stage (lines (12) and (16)). In the third case, all the remaining chunks are downloaded from the source peer $i$ such that $C_i$ is the highest capacity among the $n$ low-capacity source peers (lines (18)–(19)).

Notice that algorithm $\mathbb{HP}$ does not guarantee reduction of download time. For instance, if the threshold $c^*$ is too large, it will take long time to find a high-capacity peer, and the total download time will be longer (see Section 5). Furthermore, the order of probing also has strong impact on performance (see Section 6). If the above two problems are not solved properly, the probing algorithm does not necessarily reduce download time.

### 3.2.2. *Analysis*

Let $p_{L,i}$ denote the probability that source peer $i$ is a low-capacity source peer, i.e.,

$$p_{L,i} = \int_0^{c^*} f_{C_i}(c)dc = F_{C_i}(c^*),$$

and $p_{H,i}$ denote the probability that source peer $i$ is a high-capacity source peer, i.e.,

$$p_{H,i} = 1 - p_{L,i} = 1 - F_{C_i}(c^*) = \int_{c^*}^{\infty} f_{C_i}(c)dc.$$

Let $T_{L,i}(S)$ be the download time of a file of size $S$ from a low-capacity source peer $i$. The expectation of $T_{L,i}(S)$ is

$$\boldsymbol{E}(T_{L,i}(S)) = \frac{1}{p_{L,i}} \int_0^{c^*} T(S,c) f_{C_i}(c)dc = \frac{1}{p_{L,i}} \int_0^{c^*} \frac{S}{c} f_{C_i}(c)dc = S\boldsymbol{E}(T_{L,i}(1)),$$

where

$$\boldsymbol{E}(T_{L,i}(1)) = \frac{1}{p_{L,i}} \int_0^{c^*} \frac{f_{C_i}(c)}{c} dc.$$

Let $T_{H,i}(S)$ be the download time of a file of size $S$ from a high-capacity source peer $i$. The expectation of $T_{H,i}(S)$ is

$$\boldsymbol{E}(T_{H,i}(S)) = \frac{1}{p_{H,i}} \int_{c^*}^{\infty} T(S,c) f_{C_i}(c)dc = \frac{1}{p_{H,i}} \int_{c^*}^{\infty} \frac{S}{c} f_{C_i}(c)dc = S\boldsymbol{E}(T_{H,i}(1)),$$

**Algorithm HP:** Chunk-Based Switching and Peer Selection
**Input:** A file of size $S$ with chunks $1, 2, ..., m$ and $n$ source peers $1, 2, ..., n$.
**Output:** A download schedule for the file.

$i \leftarrow 1$;
`while` ($i \leq m$ and $i \leq n$) `do`
    download chunk $i$ from source peer $i$; // probe source peer $i$
    `if` (the service capacity of source peer $i$ is at least $c^*$, i.e., $C_i \geq c^*$)
        exit the loop; // a high-capacity source peer is identified
    `else`
        $i \leftarrow i + 1$; // probe the next source peer
    `end if`;
`end do`;
`if` ($m \leq n$)
    `if` ($i < m$) // a high-capacity source peer $i$ is found
        download the chunks $i + 1$, $i + 2$, ..., $m$ from source peer $i$;
    `end if`;
`else`
    `if` ($i \leq n$) // a high-capacity source peer $i$ is found
        download the chunks $i + 1$, $i + 2$, ..., $m$ from source peer $i$;
    `else` // download from a low-capacity peer with the highest capacity
        download the chunks $n + 1$, $n + 2$, ..., $m$ from source peer $i$
            such that $C_i = \max\{C_1, C_2, ..., C_n\}$;
    `end if`;
`end if`.

Fig. 2. A chunk-based switching and peer selection algorithm.

where

$$\boldsymbol{E}(T_{H,i}(1)) = \frac{1}{p_{H,i}} \int_{c^*}^{\infty} \frac{f_{C_i}(c)}{c} dc.$$

For homogeneous source peers, the probability that a source peer is a low-capacity source peer is

$$p_L = \int_0^{c^*} f_C(c) dc = F_C(c^*),$$

and the probability that a source peer is a high-capacity source peer is

$$p_H = 1 - p_L = 1 - F_C(c^*) = \int_{c^*}^{\infty} f_C(c) dc.$$

Let $T_L(S)$ be the download time of a file of size $S$ from a low-capacity source peer.

The expectation of $T_L(S)$ is

$$\boldsymbol{E}(T_L(S)) = \frac{1}{p_L}\int_0^{c^*} T(S,c)f_C(c)dc = \frac{1}{p_L}\int_0^{c^*} \frac{S}{c}f_C(c)dc = S\boldsymbol{E}(T_L(1)),$$

where

$$\boldsymbol{E}(T_L(1)) = \frac{1}{p_L}\int_0^{c^*} \frac{f_C(c)}{c}dc.$$

Let $T_H(S)$ be the download time of a file of size $S$ from a high-capacity source peer. The expectation of $T_L(S)$ is

$$\boldsymbol{E}(T_H(S)) = \frac{1}{p_H}\int_{c^*}^{\infty} T(S,c)f_C(c)dc = \frac{1}{p_H}\int_{c^*}^{\infty} \frac{S}{c}f_C(c)dc = S\boldsymbol{E}(T_H(1)),$$

where

$$\boldsymbol{E}(T_H(1)) = \frac{1}{p_H}\int_{c^*}^{\infty} \frac{f_C(c)}{c}dc.$$

In the following, we analyze $\boldsymbol{E}(T_{\mathsf{HP}}(S))$, the expected download time of our algorithm $\mathbb{HP}$ for a file of size $S$.

First, we consider the case when $m \le n$. If a high-capacity source peer $i$, where $1 \le i \le m$, is found in line (4), which occurs with probability

$$p_{L,1}p_{L,2}\cdots p_{L,i-1}p_{H,i},$$

since the first $i-1$ probes encounter low-capacity source peers and the $i$th probe encounters a high-capacity source peer, the expected download time is

$$\boldsymbol{E}(T_{L,1}(S^*)) + \boldsymbol{E}(T_{L,2}(S^*)) + \cdots + \boldsymbol{E}(T_{L,i-1}(S^*)) + (m-i+1)\boldsymbol{E}(T_{H,i}(S^*)),$$

because the first $i-1$ chunks are downloaded from low-capacity source peers and the last $m-i+1$ chunks are downloaded from a high-capacity source peer. If a high-capacity source peer is not found by the algorithm, which occurs with probability

$$p_{L,1}p_{L,2}\cdots p_{L,m},$$

since all the $m$ probes encounter low-capacity source peers, the expected download time is

$$\boldsymbol{E}(T_{L,1}(S^*)) + \boldsymbol{E}(T_{L,2}(S^*)) + \cdots + \boldsymbol{E}(T_{L,m}(S^*)),$$

because all the $m$ chunks are downloaded from low-capacity source peers. Summarizing the above discussion, we get the expected download time of our algorithm $\mathbb{HP}$ for a file of size $S$ for this case:

$$\begin{aligned}&\boldsymbol{E}(T_{\mathsf{HP}}(S))\\ &= \sum_{i=1}^{m} p_{L,1}p_{L,2}\cdots p_{L,i-1}p_{H,i}\left(\sum_{j=1}^{i-1}\boldsymbol{E}(T_{L,j}(S^*)) + (m-i+1)\boldsymbol{E}(T_{H,i}(S^*))\right)\\ &\quad + p_{L,1}p_{L,2}\cdots p_{L,m}\sum_{j=1}^{m}\boldsymbol{E}(T_{L,j}(S^*))\end{aligned}$$

$$= S^* \left( \sum_{i=1}^{m} p_{L,1} p_{L,2} \cdots p_{L,i-1} p_{H,i} \left( \sum_{j=1}^{i-1} \boldsymbol{E}(T_{L,j}(1)) + (m-i+1)\boldsymbol{E}(T_{H,i}(1)) \right) + p_{L,1} p_{L,2} \cdots p_{L,m} \sum_{j=1}^{m} \boldsymbol{E}(T_{L,j}(1)) \right)$$
$$= S\boldsymbol{E}(T_{\mathbf{HP}}(1))\,,$$

where

$$\boldsymbol{E}(T_{\mathbf{HP}}(1)) = \frac{1}{m} \left( \sum_{i=1}^{m} p_{L,1} p_{L,2} \cdots p_{L,i-1} p_{H,i} \left( \sum_{j=1}^{i-1} \boldsymbol{E}(T_{L,j}(1)) + (m-i+1)\boldsymbol{E}(T_{H,i}(1)) \right) + p_{L,1} p_{L,2} \cdots p_{L,m} \sum_{j=1}^{m} \boldsymbol{E}(T_{L,j}(1)) \right). \quad (2)$$

For homogeneous source peers, the above equation can be simplified and a closed form expression of $\boldsymbol{E}(T_{\mathbf{HP}}(1))$ can be obtained as follows,

$$\boldsymbol{E}(T_{\mathbf{HP}}(1)) = \frac{1}{m} \left( \sum_{i=1}^{m} p_L^{i-1} p_H ((i-1)\boldsymbol{E}(T_L(1)) + (m-i+1)\boldsymbol{E}(T_H(1))) + p_L^m m \boldsymbol{E}(T_L(1)) \right)$$
$$= \frac{1}{m} \left( p_H \boldsymbol{E}(T_L(1)) \sum_{i=1}^{m} (i-1) p_L^{i-1} + p_H \boldsymbol{E}(T_H(1)) \sum_{i=1}^{m} (m-i+1) p_L^{i-1} + p_L^m m \boldsymbol{E}(T_L(1)) \right)$$
$$= \frac{1}{m} \left( p_H \boldsymbol{E}(T_L(1)) \left( \frac{p_L - p_L^m - (m-1)(1-p_L)p_L^m}{1-p_L} \right) + p_H \boldsymbol{E}(T_H(1)) \frac{1}{1-p_L} \left( m - \frac{p_L(1-p_L^m)}{1-p_L} \right) + p_L^m m \boldsymbol{E}(T_L(1)) \right)$$
$$= \frac{1}{m} \left( (p_L - p_L^m - (m-1)(1-p_L)p_L^m + m p_L^m) \boldsymbol{E}(T_L(1)) + \left( m - \frac{p_L(1-p_L^m)}{1-p_L} \right) \boldsymbol{E}(T_H(1)) \right)$$
$$= \frac{1}{m} \left( p_L (1 + (m-1) p_L^m) \boldsymbol{E}(T_L(1)) + \left( m - \frac{p_L(1-p_L^m)}{1-p_L} \right) \boldsymbol{E}(T_H(1)) \right). \quad (3)$$

Next, we consider the case when $m > n$. If a high-capacity source peer $i$, where $1 \le i \le n$, is found in line (4), which occurs with probability

$$p_{L,1}p_{L,2}\cdots p_{L,i-1}p_{H,i},$$

the expected download time is

$$\boldsymbol{E}(T_{L,1}(S^*)) + \boldsymbol{E}(T_{L,2}(S^*)) + \cdots + \boldsymbol{E}(T_{L,i-1}(S^*)) + (m-i+1)\boldsymbol{E}(T_{H,i}(S^*)).$$

If a high-capacity source peer is not found by the algorithm, which occurs with probability

$$p_{L,1}p_{L,2}\cdots p_{L,n},$$

the expected download time is

$$\boldsymbol{E}(T_{L,1}(S^*)) + \boldsymbol{E}(T_{L,2}(S^*)) + \cdots + \boldsymbol{E}(T_{L,n}(S^*)) + (m-n)\boldsymbol{E}(T_M(S^*)),$$

where $T_M(S)$ is the download time of a file of size $S$ from the source peer that has the maximum service capacity among the $n$ low-capacity source peers (lines (18)–(19)). Hence, we get the expected download time of our algorithm HP for a file of size $S$,

$$\begin{aligned}
&\boldsymbol{E}(T_{\mathsf{HP}}(S))\\
&\quad= \sum_{i=1}^{n} p_{L,1}p_{L,2}\cdots p_{L,i-1}p_{H,i}\left(\sum_{j=1}^{i-1}\boldsymbol{E}(T_{L,j}(S^*)) + (m-i+1)\boldsymbol{E}(T_{H,i}(S^*))\right)\\
&\qquad + p_{L,1}p_{L,2}\cdots p_{L,n}\left(\sum_{j=1}^{n}\boldsymbol{E}(T_{L,j}(S^*)) + (m-n)\boldsymbol{E}(T_M(S^*))\right)\\
&\quad= S^*\left(\sum_{i=1}^{n} p_{L,1}p_{L,2}\cdots p_{L,i-1}p_{H,i}\left(\sum_{j=1}^{i-1}\boldsymbol{E}(T_{L,j}(1)) + (m-i+1)\boldsymbol{E}(T_{H,i}(1))\right)\right.\\
&\qquad \left. + p_{L,1}p_{L,2}\cdots p_{L,n}\left(\sum_{j=1}^{n}\boldsymbol{E}(T_{L,j}(1)) + (m-n)\boldsymbol{E}(T_M(1))\right)\right)\\
&\quad= S\boldsymbol{E}(T_{\mathsf{HP}}(1)),
\end{aligned}$$

where

$$\begin{aligned}
&\boldsymbol{E}(T_{\mathsf{HP}}(1))\\
&\quad= \frac{1}{m}\left(\sum_{i=1}^{n} p_{L,1}p_{L,2}\cdots p_{L,i-1}p_{H,i}\left(\sum_{j=1}^{i-1}\boldsymbol{E}(T_{L,j}(1)) + (m-i+1)\boldsymbol{E}(T_{H,i}(1))\right)\right.\\
&\qquad \left. + p_{L,1}p_{L,2}\cdots p_{L,n}\left(\sum_{j=1}^{n}\boldsymbol{E}(T_{L,j}(1)) + (m-n)\boldsymbol{E}(T_M(1))\right)\right).
\end{aligned} \tag{4}$$

Notice that as $m \to \infty$, we have

$$\begin{aligned}\boldsymbol{E}(T_{\mathsf{HP}}(1)) = &\sum_{i=1}^{n} p_{L,1}p_{L,2}\cdots p_{L,i-1}p_{H,i}\boldsymbol{E}(T_{H,i}(1))\\ &+ p_{L,1}p_{L,2}\cdots p_{L,n}\boldsymbol{E}(T_M(1)).\end{aligned} \tag{5}$$

For homogeneous source peers, the above equation can be simplified and a closed form expression of $\boldsymbol{E}(T_{\mathsf{HP}}(1))$ can be obtained as follows,

$$\begin{aligned}\boldsymbol{E}(T_{\mathsf{HP}}(1)) &= \frac{1}{m}\left(\sum_{i=1}^{n} p_L^{i-1}p_H((i-1)\boldsymbol{E}(T_L(1)) + (m-i+1)\boldsymbol{E}(T_H(1)))\right.\\ &\quad \left. + p_L^n(n\boldsymbol{E}(T_L(1)) + (m-n)\boldsymbol{E}(T_M(1)))\right)\\ &= \frac{1}{m}\left(p_H\boldsymbol{E}(T_L(1))\sum_{i=1}^{n}(i-1)p_L^{i-1} + p_H\boldsymbol{E}(T_H(1))\sum_{i=1}^{n}(m-i+1)p_L^{i-1}\right.\\ &\quad \left. + p_L^n n\boldsymbol{E}(T_L(1)) + p_L^n(m-n)\boldsymbol{E}(T_M(1))\right)\\ &= \frac{1}{m}\left(p_H\boldsymbol{E}(T_L(1))\left(\frac{p_L - p_L^n - (n-1)(1-p_L)p_L^n}{1-p_L}\right)\right.\\ &\quad + p_H\boldsymbol{E}(T_H(1))\frac{1}{1-p_L}\left(m - \frac{p_L(1-p_L^{n-1})}{1-p_L} - (m-n+1)p_L^n\right)\\ &\quad \left. + p_L^n n\boldsymbol{E}(T_L(1)) + p_L^n(m-n)\boldsymbol{E}(T_M(1))\right)\\ &= \frac{1}{m}\left((p_L - p_L^n - (n-1)(1-p_L)p_L^n + np_L^n)\boldsymbol{E}(T_L(1))\right.\\ &\quad + \left(m - \frac{p_L(1-p_L^{n-1})}{1-p_L} - (m-n+1)p_L^n\right)\boldsymbol{E}(T_H(1))\\ &\quad \left. + p_L^n(m-n)\boldsymbol{E}(T_M(1))\right)\\ &= \frac{1}{m}\left(p_L(1+(n-1)p_L^n)\boldsymbol{E}(T_L(1))\right.\\ &\quad + \left(m - \frac{p_L(1-p_L^{n-1})}{1-p_L} - (m-n+1)p_L^n\right)\boldsymbol{E}(T_H(1))\\ &\quad \left. + p_L^n(m-n)\boldsymbol{E}(T_M(1))\right).\end{aligned} \tag{6}$$

To complete the analysis, we need to find $\boldsymbol{E}(T_M(S))$. Let

$$C_M = \max\{C'_1, C'_2, ..., C'_n\},$$

where $C'_1, C'_2, ..., C'_n$ are service capacities of $n$ low-capacity source peers with pdf

$$f_{C'_i}(c) = \frac{1}{p_{L,i}} f_{C_i}(c),$$

and cdf

$$F_{C'_i}(c) = \frac{1}{p_{L,i}} F_{C_i}(c),$$

in $[0, c^*)$. The cdf of $C_M$ is

$$\begin{aligned} F_{C_M}(c) &= \boldsymbol{P}[C_M \leq c] \\ &= \prod_{i=1}^{n} \boldsymbol{P}[C'_i \leq c] \\ &= \prod_{i=1}^{n} F_{C'_i}(c) \\ &= \prod_{i=1}^{n} \frac{1}{p_{L,i}} F_{C_i}(c) \\ &= \left( \prod_{i=1}^{n} \frac{1}{p_{L,i}} \right) \left( \prod_{i=1}^{n} F_{C_i}(c) \right), \end{aligned}$$

and the pdf of $C_M$ is

$$\begin{aligned} f_{C_M}(c) &= \left( \prod_{i=1}^{n} \frac{1}{p_{L,i}} \right) \left( \sum_{i=1}^{n} f_{C_i}(c) \prod_{i' \neq i} F_{C_{i'}}(c) \right) \\ &= \left( \prod_{i=1}^{n} \frac{1}{p_{L,i}} \right) \left( \prod_{i=1}^{n} F_{C_i}(c) \right) \sum_{i=1}^{n} \frac{f_{C_i}(c)}{F_{C_i}(c)} \\ &= F_{C_M}(c) \sum_{i=1}^{n} \frac{f_{C_i}(c)}{F_{C_i}(c)}, \end{aligned}$$

for all $0 \leq c < c^*$. Thus, we get the cdf of $T_M(S) = S/C_M$,

$$\begin{aligned} F_{T_M(S)}(t) &= \boldsymbol{P}[T_M(S) \leq t] \\ &= \boldsymbol{P}\left[ \frac{S}{C_M} \leq t \right] \\ &= \boldsymbol{P}\left[ C_M \geq \frac{S}{t} \right] \end{aligned}$$

$$= 1 - F_{C_M}\left(\frac{S}{t}\right)$$
$$= 1 - \left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right)\left(\prod_{i=1}^{n} F_{C_i}\left(\frac{S}{t}\right)\right),$$

and the pdf of $T_M(S)$,

$$f_{T_M(S)}(t) = \frac{S}{t^2}\left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right)\left(\sum_{i=1}^{n} f_{C_i}\left(\frac{S}{t}\right)\prod_{i'\neq i} F_{C_{i'}}\left(\frac{S}{t}\right)\right)$$
$$= \frac{S}{t^2}\left(1 - F_{T_M(S)}(t)\right)\sum_{i=1}^{n} \frac{f_{C_i}(S/t)}{F_{C_i}(S/t)},$$

for all $t > S/c^*$. Consequently, the expectation of $T_M(S)$ is either

$$\boldsymbol{E}(T_M(S)) = \int_{S/c^*}^{\infty} t f_{T_M(S)}(t)dt,$$

or, equivalently,

$$\boldsymbol{E}(T_M(S)) = \int_0^{\infty} \left(1 - F_{T_M(S)}(t)\right)dt,$$

or, by randomizing $c$ in $T_M(S) = S/c$,

$$\boldsymbol{E}(T_M(S)) = \int_0^{c^*} T(S,c) f_{C_M}(c)dc = \int_0^{c^*} \frac{S}{c} f_{C_M}(c)dc,$$

that is,

$$\begin{aligned}
\boldsymbol{E}(T_M(S)) &= \int_0^{\infty} \left(1 - F_{T_M(S)}(t)\right)dt \\
&= \frac{S}{c^*} + \int_{S/c^*}^{\infty} \left(1 - F_{T_M(S)}(t)\right)dt \\
&= \frac{S}{c^*} + \int_{S/c^*}^{\infty} \left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right)\left(\prod_{i=1}^{n} F_{C_i}\left(\frac{S}{t}\right)\right)dt \\
&= \frac{S}{c^*} + \left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right)\int_{c^*}^{0} \left(\prod_{i=1}^{n} F_{C_i}(c)\right) d\left(\frac{S}{c}\right) \quad \left(\text{by letting } c = \frac{S}{t}\right) \\
&= \frac{S}{c^*} - \left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right)\int_0^{c^*} \left(\prod_{i=1}^{n} F_{C_i}(c)\right) d\left(\frac{S}{c}\right) \\
&= \frac{S}{c^*} - \left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right)\left(\prod_{i=1}^{n} F_{C_i}(c)\right)\left(\frac{S}{c}\right)\Bigg|_0^{c^*} \\
&\quad + \left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right)\int_0^{c^*} \frac{S}{c}\left(\sum_{i=1}^{n} f_{C_i}(c)\prod_{i'\neq i} F_{C_{i'}}(c)\right)dc
\end{aligned}$$

$$
\begin{aligned}
&= \frac{S}{c^*} - \left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right) \left(\prod_{i=1}^{n} F_{C_i}(c^*)\right) \left(\frac{S}{c^*}\right) \\
&\quad + \int_0^{c^*} \frac{S}{c} \left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right) \left(\sum_{i=1}^{n} f_{C_i}(c) \prod_{i' \neq i} F_{C_{i'}}(c)\right) dc \\
&= \int_0^{c^*} \frac{S}{c} f_{C_M}(c) dc \\
&= S\boldsymbol{E}(T_M(1)),
\end{aligned}
$$

where we notice that

$$
\left(\prod_{i=1}^{n} \frac{1}{p_{L,i}}\right) \left(\prod_{i=1}^{n} F_{C_i}(c^*)\right) = F_{C_M}(c^*) = 1,
$$

and

$$
\boldsymbol{E}(T_M(1)) = \int_0^{c^*} \frac{1}{c} f_{C_M}(c) dc = \int_0^{c^*} \frac{1}{c} F_{C_M}(c) \sum_{i=1}^{n} \frac{f_{C_i}(c)}{F_{C_i}(c)} dc. \tag{7}
$$

For homogeneous source peers, we get the cdf of $C_M$,

$$
F_{C_M}(c) = \left(\frac{F_C(c)}{p_L}\right)^n,
$$

and the pdf of $C_M$ is

$$
f_{C_M}(c) = \frac{n}{p_L} \left(\frac{F_C(c)}{p_L}\right)^{n-1} f_C(c) = \frac{n}{p_L^n} (F_C(c))^{n-1} f_C(c),
$$

for all $0 \le c < c^*$. The cdf of $T_M(S)$ is,

$$
F_{T_M(S)}(t) = 1 - \left(\frac{1}{p_L} F_C\left(\frac{S}{t}\right)\right)^n,
$$

and the pdf of $T_M(S)$ is,

$$
f_{T_M(S)}(t) = \frac{nS}{p_L^n t^2} \left(F_C\left(\frac{S}{t}\right)\right)^{n-1} f_C\left(\frac{S}{t}\right),
$$

for all $t > S/c^*$. Consequently, the expectation of $T_M(S)$ is

$$
\boldsymbol{E}(T_M(S)) = \int_0^{c^*} T(S,c) f_{C_M}(c) dc = \int_0^{c^*} \frac{S}{c} \cdot \frac{n}{p_L^n} (F_C(c))^{n-1} f_C(c) dc,
$$

by randomizing $c$ in $T_M(S) = S/c$. We can also represent $\boldsymbol{E}(T_M(S))$ as

$$
\boldsymbol{E}(T_M(S)) = S\boldsymbol{E}(T_M(1)),
$$

where

$$
\boldsymbol{E}(T_M(1)) = \frac{n}{p_L^n} \int_0^{c^*} \frac{1}{c} (F_C(c))^{n-1} f_C(c) dc. \tag{8}
$$

## 4. Performance Comparison

In this section, we present a numerical example to compare the performance of algorithms $\mathbb{RP}$ and $\mathbb{HP}$. As in most P2P file sharing and exchange systems, the file sizes $S$ are in the range $10 \sim 1500$ MB [1]. We set the chunk size $S^* = 10$ MB. This implies that the number $m$ of chunks is in the range $1 \sim 150$. 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 $n = 10$ source peers. Assume that $C_i$ has a uniform distribution in $[c_{i,1}, c_{i,2}]$, where $c_{i,1} = 3.7 - 0.1i$ and $c_{i,2} = 4 + 0.2i$, for all $1 \le i \le n$, i.e.,

$$f_{C_i}(c) = \frac{1}{c_{i,2} - c_{i,1}},$$

and

$$F_{C_i}(c) = \frac{c - c_{i,1}}{c_{i,2} - c_{i,1}},$$

for all $c_{i,1} \le c \le c_{i,2}$. By straightforward calculation, we obtain

$$\begin{aligned}
p_{L,i} &= \frac{c^* - c_{i,1}}{c_{i,2} - c_{i,1}}, \\
p_{H,i} &= \frac{c_{i,2} - c^*}{c_{i,2} - c_{i,1}}, \\
\boldsymbol{E}(T_{L,i}(1)) &= \frac{\ln(c^*/c_{i,1})}{c^* - c_{i,1}}, \\
\boldsymbol{E}(T_{H,i}(1)) &= \frac{\ln(c_{i,2}/c^*)}{c_{i,2} - c^*}, \\
\boldsymbol{E}(T_M(1)) &= \int_0^{c^*} \frac{1}{c} F_{C_M}(c) \sum_{i=1}^{n} \frac{f_{C_i}(c)}{F_{C_i}(c)} dc \quad \text{(by Eq. (7))} \\
&= \int_{c_1}^{c^*} \frac{1}{c} \left( \prod_{i=1}^{n} \frac{1}{p_{L,i}} \right) \left( \prod_{i=1}^{n} F_{C_i}(c) \right) \sum_{i=1}^{n} \frac{f_{C_i}(c)}{F_{C_i}(c)} dc\,,
\end{aligned}$$

where $c_1 = \max\{c_{1,1}, c_{2,1}, ..., c_{n,1}\}$, since we must have $C_M \ge \max\{c_{1,1}, c_{2,1}, ..., c_{n,1}\}$.

In Figure 3, we demonstrate the expected download time $\boldsymbol{E}(T_{\mathbb{HP}}(S))$ of algorithm $\mathbb{HP}$ for a file of size $S = mS^*$, where $0 \le m \le 100$. The service capacity threshold is set as $c^* = 3.7, 3.9, 4.1$, which are chosen to illustrate our key observations. We also show the expected download time $\boldsymbol{E}(T_{\mathbb{RP}}(S))$ of algorithm $\mathbb{RP}$. As we have known already in Eq. (5), as $m$ increases, $\boldsymbol{E}(T_{\mathbb{HP}}(S))$ approaches a linear function with slope

$$\sum_{i=1}^{n} p_{L,1} p_{L,2} \cdots p_{L,i-1} p_{H,i} \boldsymbol{E}(T_{H,i}(1)) + p_{L,1} p_{L,2} \cdots p_{L,n} \boldsymbol{E}(T_M(1)),$$

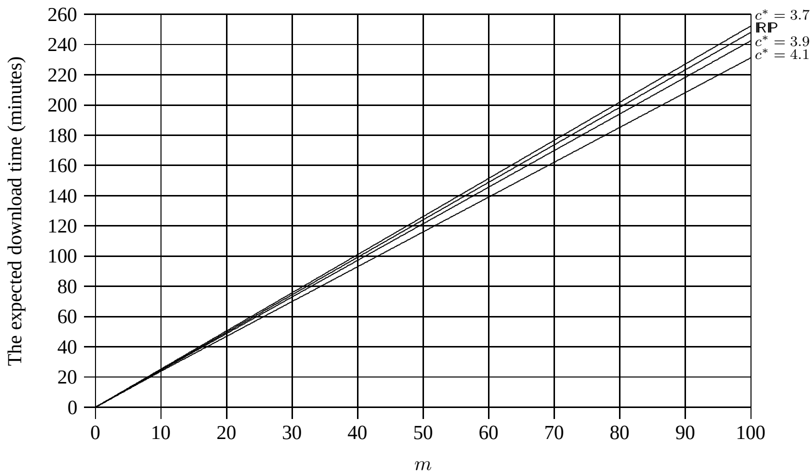


Fig. 3. The expected download time versus file size.

which is determined by two factors, namely, the service capacity threshold and the order of source peers. It is clear that the $p_{L,i}$'s, the $p_{H,i}$'s, and the $\boldsymbol{E}(T_{H,i}(1))$'s depend on $c^*$, and the above slope further depends on the order of probing. For a poorly chosen $c^*$ and a poorly chosen order of probing, algorithm $\mathbb{HP}$ can perform worse than algorithm $\mathbb{RP}$. For instance, when $c^* = 3.7$ and the source peers are arranged in the decreasing order of the $\boldsymbol{E}(T_{H,i}(1))$'s as we have done in Figure 3, that is,

$$\boldsymbol{E}(T_{H,1}(1)) \geq \boldsymbol{E}(T_{H,2}(1)) \geq \cdots \geq \boldsymbol{E}(T_{H,n}(1)),$$

we have $\boldsymbol{E}(T_{\mathbb{HP}}(S)) > \boldsymbol{E}(T_{\mathbb{RP}}(S))$. However, if we set $c^* = 3.9$ or $4.1$ as we have done in Figure 3, we get $\boldsymbol{E}(T_{\mathbb{HP}}(S)) < \boldsymbol{E}(T_{\mathbb{RP}}(S))$. As shown later, the performance of algorithm $\mathbb{HP}$ can be significantly improved if the source peers are arranged in the increasing order of the $\boldsymbol{E}(T_{H,i}(1))$'s.

## 5. Optimal Threshold of Service Capacity

When $c^* \to 0$, we have $p_{L,i} \to 0$ and $p_{H,i} \to 1$, for all $1 \leq i \leq n$. This essentially means that algorithm $\mathbb{HP}$ simply downloads a file from the first source peer without much consideration. This extreme case certainly does not work well. When $c^* \to \infty$, we have $p_{L,i} \to 1$ and $p_{H,i} \to 0$, for all $1 \leq i \leq n$. This essentially means that algorithm $\mathbb{HP}$ probes all $n$ source peers and chooses the one with the largest service capacity. This extreme case works great if $m$ is large, such that the time for probing in the first stage is negligible. However, it may not work well for small to moderate sized files.

The selection of the service capacity threshold $c^*$ has strong impact on the performance of algorithm $\mathbb{HP}$. If $c^*$ is too small, a high-capacity source peer may be identified too quickly; however, the service capacity of the so called "high-capacity" source peer may not be high. If $c^*$ is too large, the time required to find a high-capacity source peer may be too long, and by the time a high-capacity source peer is identified, a file may be almost downloaded. Thus, there is an optimal choice of $c^*$ which minimizes $\boldsymbol{E}(T_{\mathsf{HP}}(S))$ (actually, $\boldsymbol{E}(T_{\mathsf{HP}}(1))$). Notice that the optimal value of $c^*$ is independent of $S$.

Consider the case when source peers are homogeneous. Assume that $C$ has a uniform distribution in $[c_1, c_2]$, i.e.,

$$f_C(c) = \frac{1}{c_2 - c_1},$$

and

$$F_C(c) = \frac{c - c_1}{c_2 - c_1},$$

for all $c_1 \le c \le c_2$. By straightforward calculation, we obtain

$$\begin{aligned}
p_L &= \frac{c^* - c_1}{c_2 - c_1}, \\
p_H &= \frac{c_2 - c^*}{c_2 - c_1}, \\
\boldsymbol{E}(T_L(1)) &= \frac{\ln(c^*/c_1)}{c^* - c_1}, \\
\boldsymbol{E}(T_H(1)) &= \frac{\ln(c_2/c^*)}{c_2 - c^*}, \\
\boldsymbol{E}(T_M(1)) &= \frac{n}{p_L^n} \int_{c_1}^{c^*} \frac{1}{c} \left(F_C(c)\right)^{n-1} f_C(c) dc \quad \text{(by Eq. (8))} \\
&= n \left(\frac{c_2 - c_1}{c^* - c_1}\right)^n \int_{c_1}^{c^*} \frac{1}{c} \left(\frac{c - c_1}{c_2 - c_1}\right)^{n-1} \frac{1}{c_2 - c_1} dc \\
&= \frac{n}{(c^* - c_1)^n} \int_{c_1}^{c^*} \frac{(c - c_1)^{n-1}}{c} dc \\
&= \frac{n}{(c^* - c_1)^n} \int_{c_1}^{c^*} \frac{1}{c} \sum_{j=0}^{n-1} \binom{n-1}{j} c^j (-c_1)^{n-j-1} dc \\
&= \frac{n}{(c^* - c_1)^n} \int_{c_1}^{c^*} \frac{1}{c} \left( (-c_1)^{n-1} + \sum_{j=1}^{n-1} \binom{n-1}{j} c^j (-c_1)^{n-j-1} \right) dc \\
&= \frac{n}{(c^* - c_1)^n} \left( (-c_1)^{n-1} \int_{c_1}^{c^*} \frac{dc}{c} + \sum_{j=1}^{n-1} \binom{n-1}{j} (-c_1)^{n-j-1} \int_{c_1}^{c^*} c^{j-1} dc \right) \\
&= \frac{n}{(c^* - c_1)^n} \left( (-c_1)^{n-1} \ln\left(\frac{c^*}{c_1}\right) + \sum_{j=1}^{n-1} \binom{n-1}{j} (-c_1)^{n-j-1} \frac{(c^*)^j - c_1^j}{j} \right).
\end{aligned}$$

Therefore, by Eq. (6), we obtain a closed form expression of $\boldsymbol{E}(T_{\mathsf{HP}}(1))$,

$$
\begin{aligned}
\boldsymbol{E}(T_{\mathsf{HP}}(1)) &= \frac{1}{m}\Bigg(p_L(1+(n-1)p_L^n)\boldsymbol{E}(T_L(1)) \\
&\quad + \left(m - \frac{p_L(1-p_L^{n-1})}{1-p_L} - (m-n+1)p_L^n\right)\boldsymbol{E}(T_H(1)) \\
&\quad + p_L^n(m-n)\boldsymbol{E}(T_M(1))\Bigg) \\
&= \frac{1}{m}\left(\frac{c^*-c_1}{c_2-c_1}\left(1+(n-1)\left(\frac{c^*-c_1}{c_2-c_1}\right)^n\right)\frac{\ln(c^*/c_1)}{c^*-c_1}\right. \\
&\quad + \left(m - \frac{c^*-c_1}{c_2-c^*}\left(1-\left(\frac{c^*-c_1}{c_2-c_1}\right)^{n-1}\right)\right. \\
&\quad \left. - (m-n+1)\left(\frac{c^*-c_1}{c_2-c_1}\right)^n\right)\frac{\ln(c_2/c^*)}{c_2-c^*} \\
&\quad + \left(\frac{c^*-c_1}{c_2-c_1}\right)^n \frac{(m-n)n}{(c^*-c_1)^n}\left((-c_1)^{n-1}\ln\left(\frac{c^*}{c_1}\right)\right. \\
&\quad \left.\left. + \sum_{j=1}^{n-1}\binom{n-1}{j}(-c_1)^{n-j-1}\frac{(c^*)^j - c_1^j}{j}\right)\right). \qquad (9)
\end{aligned}
$$

Let us consider a P2P file sharing system with $n = 10$ source peers. The uniform distribution is in the range $[c_1, c_2]$ with $c_1 = 0.375$ and $c_2 = 7.5$. In Figure 4, we demonstrate the expected download time $\boldsymbol{E}(T_{\mathsf{HP}}(S))$ of algorithm $\mathbb{HP}$ as a function of $c^*$ for a file of size $S = mS^*$, where $m = 20, 40, 60, 80, 100$. It is observed that the service capacity threshold has strong impact on the expected download time. There is a noticeable range of the expected download time as $c^*$ varies. As $c^*$ increases, $\boldsymbol{E}(T_{\mathsf{HP}}(S))$ decreases significantly; however, beyond certain point, $\boldsymbol{E}(T_{\mathsf{HP}}(S))$ starts to increase, i.e., the performance of algorithm $\mathbb{HP}$ gets worse. There is clearly an optimal choice of $c^*$ that optimizes the performance of algorithm $\mathbb{HP}$.

In Figures 5 and 6, we demonstrate the expected download time $\boldsymbol{E}(T_{\mathsf{HP}}(S))$ of algorithm $\mathbb{HP}$ for a file of size $S = 10S^*$ and $S = 100S^*$ respectively. The number of source peers is $n = 2, 3, 4, 5, 6, 10$. It is observed that even $n = 2$ source peers significantly reduce $\boldsymbol{E}(T_{\mathsf{HP}}(S))$, as compared with $\boldsymbol{E}(T_{\mathsf{HP}}(S)) = 42$ and 420 respectively when there is $n = 1$ source peer. It is also observed (and intuitively acceptable) that with a reasonable choice of $c^*$, more source peers further improve the performance of algorithm $\mathbb{HP}$, since the chance to probe a high-capacity source peer increases with $n$, that is, the probability that all source peers are low-capacity source peers vanishes as $n$ gets large. Again, for a fixed $n$, there is an optimal choice of $c^*$ that minimizes $\boldsymbol{E}(T_{\mathsf{HP}}(S))$. It is a little surprising at the first glance

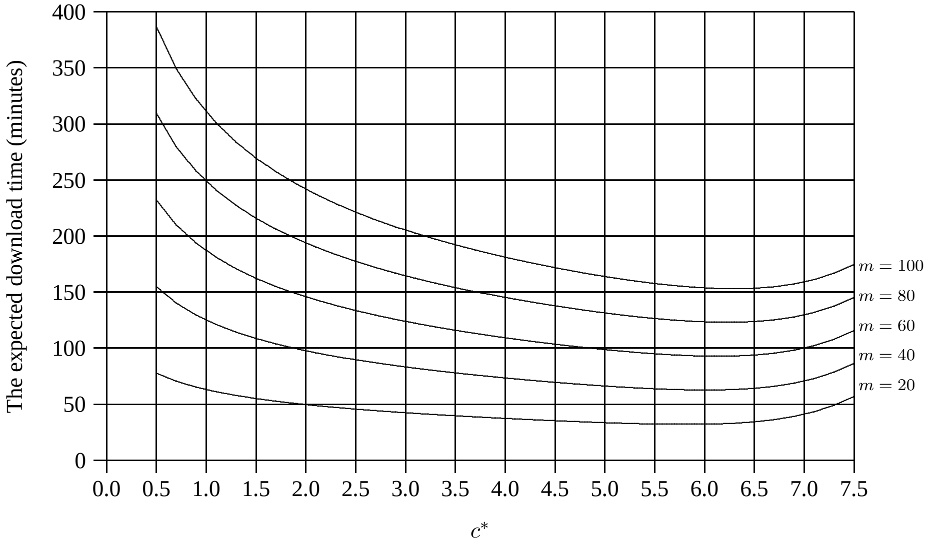


Fig. 4. The expected download time versus service capacity threshold ($n = 10$).

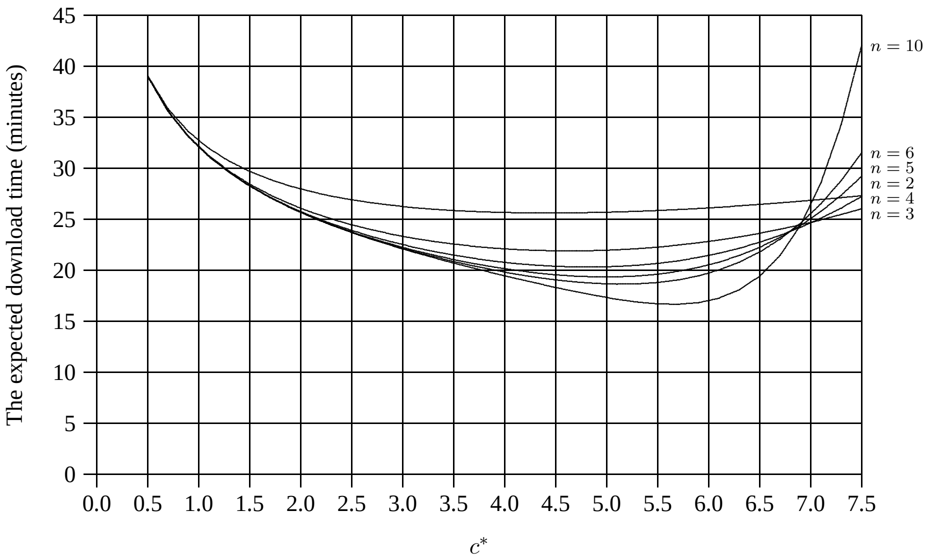


Fig. 5. The expected download time versus service capacity threshold ($m = 10$).

that when $c^*$ exceeds certain limit, more source peers degrade the performance of algorithm $\mathbb{HP}$. For instance, for $m = 10$ in Figure 5, for $c^*$ close to $c_2$, $\boldsymbol{E}(T_{\mathbb{HP}}(S))$ increases in the order of $n = 3, 4, 2, 5, 6, 10$, that is, the performance of algorithm $\mathbb{HP}$ is worse when $n = 4$ than $n = 3$. When $n = 5, 6, 10$, the performance is even

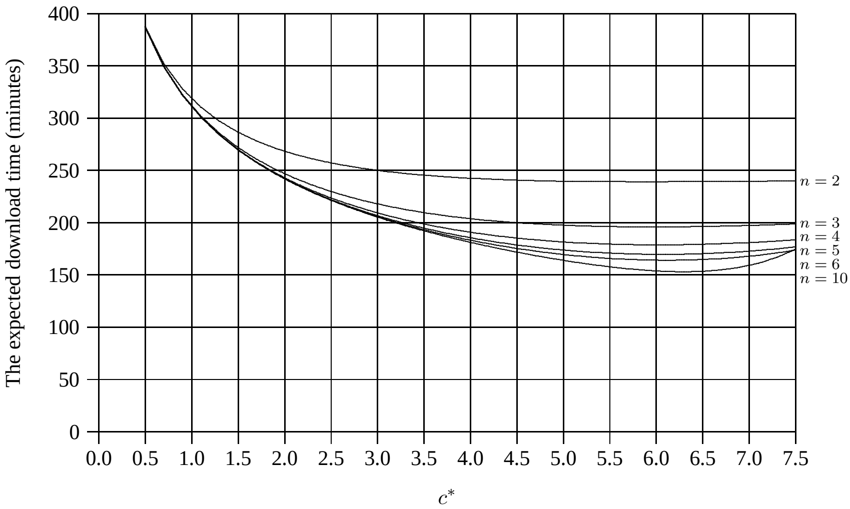


Fig. 6. The expected download time versus service capacity threshold ($m = 100$).

worse than when $n = 2$. In fact, this is not difficult to explain. When $m = 10$, i.e., there are not many chunks, the time spent for probing source peers is a significant part of the overall download time. When $c^*$ is very large, the chance to encounter a high-capacity source peers decreases, since all source peers are low-capacity source peers. Hence, algorithm $\mathbb{HP}$ essentially becomes algorithm $\mathbb{RP}$. It would be better to stop probing earlier and to choose a source peer with the largest capacity to finish the download. (Notice that the curves collapse when $c^*$ is very small (actually, at $c^* = c_1 = 0.375$). The reason is that when $c^*$ is too small, the probing process will terminate very soon, since virtually all source peers are high-capacity peers. In other words, when $c^*$ is too small, the time to identify a high-capacity peer is almost independent of $n$.)

We observe from Figures 5 and 6 that the optimal choice of the service capacity threshold increases as $m$ and $n$ increase. For a fixed $m$, the optimal choice of the service capacity threshold is relatively stable and increases with $n$ very slowly. Therefore, it is very informative to find the optimal choice of the service capacity threshold when $n = 2$. Our main result of this section is the following theorem.

**Theorem 1.** *An optimal choice of the service capacity threshold is the unique solution $c^*$ of the following equation,*

$$2\left(\frac{c^* - c_1}{(c_2 - c_1)^2}\right)\ln\left(\frac{c^*}{c_1}\right) + \left(1 + \left(\frac{c^* - c_1}{c_2 - c_1}\right)^2\right)\frac{1}{c^*}$$
$$+\left(\frac{m-1}{c_2 - c_1}\right)\ln\left(\frac{c_2}{c^*}\right) - \left((m-1)\left(\frac{c^* - c_1}{c_2 - c_1}\right) + m\right)\frac{1}{c^*} + \frac{2(m-2)}{c_2 - c_1}\left(1 - \frac{c_1}{c^*}\right) = 0,$$

*for $n = 2$ homogeneous source peers whose service capacity has a uniform distribution in the interval $[c_1, c_2]$.*

*Proof.* Based on our previous discussion, when $n = 2$, by Eq. (9), we have

$$\begin{aligned}
\boldsymbol{E}(T_{\mathsf{HP}}(1)) &= \frac{1}{m}\left(\frac{1}{c_2-c_1}\left(1+\left(\frac{c^*-c_1}{c_2-c_1}\right)^2\right)\ln\left(\frac{c^*}{c_1}\right)\right. \\
&\quad + \left(m-\frac{c^*-c_1}{c_2-c_1}-(m-1)\left(\frac{c^*-c_1}{c_2-c_1}\right)^2\right)\frac{\ln(c_2/c^*)}{c_2-c^*} \\
&\quad \left. + \frac{2(m-2)}{(c_2-c_1)^2}\left(c^*-c_1-c_1\ln\left(\frac{c^*}{c_1}\right)\right)\right) \\
&= \frac{1}{m}\left(\frac{1}{c_2-c_1}\left(1+\left(\frac{c^*-c_1}{c_2-c_1}\right)^2\right)\ln\left(\frac{c^*}{c_1}\right)\right. \\
&\quad + \left(1-\frac{c^*-c_1}{c_2-c_1}\right)\left((m-1)\left(\frac{c^*-c_1}{c_2-c_1}\right)+m\right)\frac{\ln(c_2/c^*)}{c_2-c^*} \\
&\quad \left. + \frac{2(m-2)}{(c_2-c_1)^2}\left(c^*-c_1-c_1\ln\left(\frac{c^*}{c_1}\right)\right)\right) \\
&= \frac{1}{m}\left(\frac{1}{c_2-c_1}\left(1+\left(\frac{c^*-c_1}{c_2-c_1}\right)^2\right)\ln\left(\frac{c^*}{c_1}\right)\right. \\
&\quad + \frac{1}{c_2-c_1}\left((m-1)\left(\frac{c^*-c_1}{c_2-c_1}\right)+m\right)\ln\left(\frac{c_2}{c^*}\right) \\
&\quad \left. + \frac{2(m-2)}{(c_2-c_1)^2}\left(c^*-c_1-c_1\ln\left(\frac{c^*}{c_1}\right)\right)\right) \\
&= \frac{y(c^*)}{m(c_2-c_1)},
\end{aligned}$$

where

$$\begin{aligned}
y(c^*) = &\left(1+\left(\frac{c^*-c_1}{c_2-c_1}\right)^2\right)\ln\left(\frac{c^*}{c_1}\right) \\
&+ \left((m-1)\left(\frac{c^*-c_1}{c_2-c_1}\right)+m\right)\ln\left(\frac{c_2}{c^*}\right) \\
&+ \frac{2(m-2)}{c_2-c_1}\left(c^*-c_1-c_1\ln\left(\frac{c^*}{c_1}\right)\right).
\end{aligned}$$

Table 1. Optimal service capacity threshold ($n = 2$).

| $m$ | $c^*$ |
|---|---|
| 10 | 4.4290461 |
| 20 | 4.8968135 |
| 30 | 5.1645373 |
| 40 | 5.3493051 |
| 50 | 5.4887148 |
| 60 | 5.5996561 |
| 70 | 5.6911521 |
| 80 | 5.7685772 |
| 90 | 5.8353829 |
| 100 | 5.8939128 |
| 110 | 5.9458274 |
| 120 | 5.9923445 |
| 130 | 6.0343820 |
| 140 | 6.0726481 |
| 150 | 6.1076999 |

To minimize $\boldsymbol{E}(T_{\mathsf{HP}}(1))$, it is equivalent to minimize $y(c^*)$. Notice that

$$\begin{aligned}\frac{\partial y(c^*)}{\partial c^*} &= 2\left(\frac{c^*-c_1}{(c_2-c_1)^2}\right)\ln\left(\frac{c^*}{c_1}\right) + \left(1+\left(\frac{c^*-c_1}{c_2-c_1}\right)^2\right)\frac{1}{c^*}\\ &\quad + \left(\frac{m-1}{c_2-c_1}\right)\ln\left(\frac{c_2}{c^*}\right) - \left((m-1)\left(\frac{c^*-c_1}{c_2-c_1}\right)+m\right)\frac{1}{c^*}\\ &\quad + \frac{2(m-2)}{c_2-c_1}\left(1-\frac{c_1}{c^*}\right).\end{aligned}$$

Thus, we only need to find $c^*$ for the equation $\partial y(c^*)/\partial c^* = 0$. Since the partial derivative $\partial y(c^*)/\partial c^*$ is an increasing function in the interval $[c_1, c_2]$, there is a unique solution $c^*$ for the equation. □

Although there is no closed-form solution, the equation $\partial y(c^*)/\partial c^* = 0$ can be solved numerically. In Table 1, we display the optimal choice of the service capacity threshold for $n = 2$ homogeneous source peers whose service capacity has a uniform distribution in $[c_1, c_2]$ with $c_1 = 0.375$ and $c_2 = 7.5$.

The optimal choice of the service capacity threshold approaches $c_2$ as $m$ and $n$ get large. Notice that for large $n$, by Eq. (6), we have

$$\begin{aligned}\boldsymbol{E}(T_{\mathsf{HP}}(1)) &\approx \frac{1}{m}\left(p_L\boldsymbol{E}(T_L(1)) + \left(m-\frac{p_L}{p_H}\right)\boldsymbol{E}(T_H(1))\right)\\ &= \frac{1}{m}\left(\left(\frac{c^*-c_1}{c_2-c_1}\right)\frac{\ln(c^*/c_1)}{c^*-c_1} + \left(m-\frac{c^*-c_1}{c_2-c^*}\right)\frac{\ln(c_2/c^*)}{c_2-c^*}\right)\\ &= \frac{1}{m}\left(\frac{\ln(c^*/c_1)}{c_2-c_1} + \frac{(c_1+mc_2-(m+1)c^*)\ln(c_2/c^*)}{(c_2-c^*)^2}\right).\end{aligned}$$

To minimize $\boldsymbol{E}(T_{\mathsf{HP}}(1))$, we need to have $\partial \boldsymbol{E}(T_{\mathsf{HP}}(1))/\partial c^* = 0$. By straightforward algebraic manipulation, we know that we need to find $c^*$ which satisfies the following equation,

$$\frac{(c_2 - c^*)^3}{(c_2 - c_1)c^*} + (2c_1 + (m-1)c_2 - (m+1)c^*) \ln \frac{c_2}{c^*} = (c_1 + mc_2 - (m+1)c^*) \left( \frac{c_2}{c^*} - 1 \right) .$$

The above equation has one solution $c^* = c_2$.

## 6. Optimal Order of Probing

The order of $n$ heterogeneous source peers in algorithm $\mathbb{HP}$ has strong impact on the performance of the algorithm, since the order of the source peers is the order of probing in algorithm $\mathbb{HP}$, and the order determines how easily and quickly a high-capacity source peer can be identified. Given $n$ source peers characterized by $f_{C_1}(c)$, $f_{C_2}(c)$, ..., $f_{C_n}(c)$, and a file of size $S$, the problem of finding an optimal order of the $n$ source peers such that the expected download time $\boldsymbol{E}(T_{\mathsf{HP}}(S))$ is minimized is a well defined optimization problem.

The main result of this section is the following theorem, which gives an optimal order when $m$ is sufficiently large.

**Theorem 2.** *When $m$ is sufficiently large, an optimal order $(j_1, j_2, ..., j_n)$ of the $n$ source peers that minimizes the expected download time $\boldsymbol{E}(T_{\mathsf{HP}}(S))$ satisfies*

$$\boldsymbol{E}(T_{H,j_1}(1)) \leq \boldsymbol{E}(T_{H,j_2}(1)) \leq \cdots \leq \boldsymbol{E}(T_{H,j_n}(1)).$$

*In other words, source peers should be probed in an increasing order of their expected download times when they are high-capacity source peers.*

*Proof.* We notice that minimizing $\boldsymbol{E}(T_{\mathsf{HP}}(S))$ is equivalent to minimizing $\boldsymbol{E}(T_{\mathsf{HP}}(1))$. As mentioned in Section 3, when $m$ is sufficiently large, by Eq. (5), $\boldsymbol{E}(T_{\mathsf{HP}}(1))$ is actually

$$\begin{aligned}\boldsymbol{E}(T_{\mathsf{HP}}(1)) &= \sum_{i=1}^{n} p_{L,1}p_{L,2}\cdots p_{L,i-1}p_{H,i}\boldsymbol{E}(T_{H,i}(1)) + p_{L,1}p_{L,2}\cdots p_{L,n}\boldsymbol{E}(T_M(1)) \\ &= p_{H,1}\boldsymbol{E}(T_{H,1}(1)) + p_{L,1}p_{H,2}\boldsymbol{E}(T_{H,2}(1)) \\ &\quad + \sum_{i=3}^{n} p_{L,1}p_{L,2}\cdots p_{L,i-1}p_{H,i}\boldsymbol{E}(T_{H,i}(1)) + p_{L,1}p_{L,2}\cdots p_{L,n}\boldsymbol{E}(T_M(1)) .\end{aligned}$$

First, we show that $\boldsymbol{E}(T_{\mathsf{HP}}(1))$ is minimized if and only if $\boldsymbol{E}(T_{H,1}(1)) \leq \boldsymbol{E}(T_{H,2}(1))$. Let us exchange the order of the first two source peers and get

$$\begin{aligned}\boldsymbol{E}'(T_{\mathsf{HP}}(1)) &= p_{H,2}\boldsymbol{E}(T_{H,2}(1)) + p_{L,2}p_{H,1}\boldsymbol{E}(T_{H,1}(1)) \\ &\quad + \sum_{i=3}^{n} p_{L,2}p_{L,1}\cdots p_{L,i-1}p_{H,i}\boldsymbol{E}(T_{H,i}(1)) + p_{L,2}p_{L,1}\cdots p_{L,n}\boldsymbol{E}(T_M(1)) .\end{aligned}$$

It is observed that except the first two terms in the right hand side of the above equation, all other terms remain the same. Hence, we get

$$\boldsymbol{E}(T_{\mathsf{HP}}(1)) - \boldsymbol{E}'(T_{\mathsf{HP}}(1)) = p_{H,1}p_{H,2}(\boldsymbol{E}(T_{H,1}(1)) - \boldsymbol{E}(T_{H,2}(1))),$$

and $\boldsymbol{E}(T_{\mathsf{HP}}(1) \le \boldsymbol{E}'(T_{\mathsf{HP}}(1))$ if and only if $\boldsymbol{E}(T_{H,1}(1)) \le \boldsymbol{E}(T_{H,2}(1))$. Next, we prove by induction on $n \ge 2$ that $\boldsymbol{E}(T_{\mathsf{HP}}(1))$ is minimized if and only if the $n$ source peers are arranged in an order $(j_1, j_2, ..., j_n)$ such that

$$\boldsymbol{E}(T_{H,j_1}(1)) \le \boldsymbol{E}(T_{H,j_2}(1)) \le \cdots \le \boldsymbol{E}(T_{H,j_n}(1)).$$

Notice that the base case when $n = 2$ has already been shown above. When $n > 2$, we have

$$\boldsymbol{E}(T_{\mathsf{HP}}(1)) = p_{H,1}\boldsymbol{E}(T_{H,1}(1)) + p_{L,1}\tilde{\boldsymbol{E}}(T_{\mathsf{HP}}(1)),$$

where $\tilde{\boldsymbol{E}}(T_{\mathsf{HP}}(1))$ is the expected download time of one unit of data from source peers $2, 3, ..., n$. By the induction hypothesis, $\tilde{\boldsymbol{E}}(T_{\mathsf{HP}}(1))$ is minimized if and only if

$$\boldsymbol{E}(T_{H,2}(1)) \le \boldsymbol{E}(T_{H,3}(1)) \le \cdots \le \boldsymbol{E}(T_{H,n}(1)).$$

Thus, $\boldsymbol{E}(T_{\mathsf{HP}}(1))$ is minimized if and only if

$$\boldsymbol{E}(T_{H,1}(1)) \le \boldsymbol{E}(T_{H,2}(1)) \le \cdots \le \boldsymbol{E}(T_{H,n}(1)).$$

The theorem is proved. □

In Figure 7, we demonstrate the expected download time $\boldsymbol{E}(T_{\mathsf{HP}}(S))$ of algorithm $\mathbb{HP}$ using exactly the same data in Figure 3. The only difference is that the source peers are arranged in the increasing order of the $\boldsymbol{E}(T_{H,i}(1))$'s, that is,

$$\boldsymbol{E}(T_{H,1}(1)) \le \boldsymbol{E}(T_{H,2}(1)) \le \cdots \le \boldsymbol{E}(T_{H,n}(1)).$$

We observe that such an optimal order of probing yields noticeable performance improvement.

## 7. Conclusions

We have addressed the problem of reducing download times in peer-to-peer file sharing systems with stochastic service capacities. We gave a precise analysis of the expected download time when the service capacity of a source peer is a random variable. We proposed a chunk-based switching and peer selection algorithm and analyzed the expected download time of the algorithm. We have solved the two subproblems of the optimal choice of the threshold of high-capacity source peers and the optimal order of probing. We compared the performance of our algorithm with the random chunk-based switching method and obtained noticeable performance improvement.

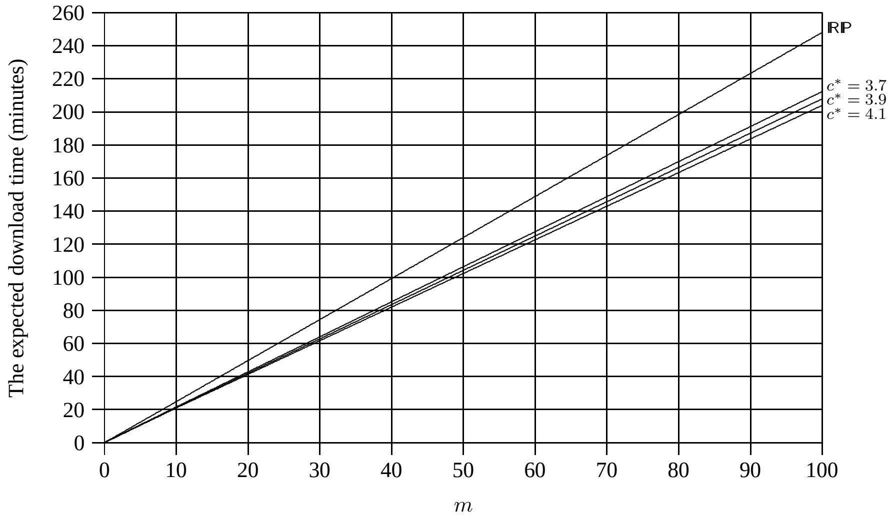


Fig. 7. The expected download time versus file size.

We would like to mention that the basis for probing is the assumption that the service capacity of a source peer does no change after probing at least for a reasonable amount of time (e.g., within the time of downloading a file). If the service capacity of a source peer changes after probing, then the technique of probing might need to be enhanced, so that it is useful and helpful in reducing the download time. It is therefore an interesting and challenging problem to propose and analyze new probing algorithms to include temporal fluctuation of source peer capacities into consideration. It is conceivable that such investigation needs significantly new insights which are well beyond the scope of this paper. In this sense, our effort in this paper is only an initial attempt towards this direction.

## Acknowledgments

The author is grateful to two reviewers who carefully read the manuscript and provided many suggestions to improve the presentation of the paper. A preliminary version of the paper was presented on the 13th Workshop on Advances in Parallel and Distributed Computational Models, Anchorage, Alaska, May 16-20, 2011.

## 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/Jensen's_inequality
[4] http://en.wikipedia.org/wiki/Peer-to-peer
[5] 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.

[6] 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.

[7] 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.

[8] J. W. Byers, M. Luby, and M. Mitzenmacher, "Accessing multiple mirror sites in parallel: using Tornado codes to speed up downloads," *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 275-283, 1999.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] C. Gkantsidis, M. Ammar, and E. Zegura, "On the effect of large-scale deployment of parallel downloading," *Proceedings of the 3rd IEEE Workshop on Internet Applications*, pp. 79-89, 2003.

[15] 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.

[16] S. G. M. Koo, C. Rosenberg, and D. Xu, "Analysis of parallel downloading for large file distribution," *Proceedings of the 9th IEEE Workshop on Future Trends of Distributed Computing Systems*, pp. 128-135, 2003.

[17] 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.

[18] K. Li, "Analysis of random time-based switching for file sharing in peer-to-peer networks," *Proceedings of the 24th IEEE International Parallel and Distributed Processing Symposium Workshops* (7th International Workshop on Hot Topics in Peer-to-Peer Systems), 2010.

[19] K. Li, "Reducing download times in peer-to-peer file sharing systems with stochastic service capacities," *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, 2011.

[20] Y. Liu, W. Gong, and P. Shenoy, "On the impact of concurrent downloads," *Proceedings of the 33nd Winter Simulation Conference*, pp. 1300-1305, 2001.

[21] 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.

[22] 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.

[23] D. Manini and M. Gribaudo, "Modelling search, availability, and parallel download in P2P file sharing applications with fluid model," *Proceedings of* 14*th International Conference on Advanced Computing and Communications*, pp. 449-454, 2006.

[24] P. Rodriguez and E. W. Biersack, "Dynamic parallel access to replicated content in the internet," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 455-465, 2002.

[25] 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* 4*th ACM International Conference on Emerging Networking Experiments and Technologies*, 2008.

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

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