Minimizing File Download Time over Stochastic Channels in Peer-to-Peer Networks

Yuh-Ming Chiu and Do Young Eun
Department of Electrical and Computer Engineering
North Carolina State University
{ychiu, dyeun}@ncsu.edu

Abstract — **The average download time of a file is an important performance metric for a user in a peer-to-peer network. We point out that the common approach of analyzing the average download time based on *average service capacity* is fundamentally flawed, and show that spatial heterogeneity and temporal correlation in the service capacity over different paths are the two major factors that have negative impact on the average file download time. We then propose a simple and distributed algorithm that can completely remove this negative impact of the two factors and yield the smallest possible average download time for each user in the network.**

I. Introduction

The P2P file-sharing applications are becoming increasingly popular and account for more than 80% of the Internet's bandwidth usage. Being a completely distributed system, the service capacity of a P2P network, or the aggregated throughput, is shown to increase as the number of uploading peers increases in the network [10, 14]. Thus, in an ideal case, the available bandwidth to each downloading peer will be constrained only by the limitation of the access link at the downloading site.

However, there are many factors that prevent a peer from fully utilizing the available bandwidth. The distribution of the data in the network, peer selection algorithms, free-riders, and many other reasons impact the performance that a downloading peer receives. It is shown [5] that, in reality, even downloading files of less than 10 MB in size may take from 5 minutes up to several hours. The downloading time for a bigger file, say, 100 MB – 1 GB, can range from hours to a whole week. Further, even when all users try to download the same file, each of them may have very different downloading times, depending on the available capacity fluctuation, the path it chooses to download the file, etc.

It is a common belief that there exist a direct relationship between the file download time and the service capacity the network can offer. The average download time of a file of size F is simply given by F/c, where c is the *average service capacity*. This approach using the average service capacity to analyze the average download time has been a common practice in the literature [8, 4, 10, 14, 1, 2, 11]. However, the service capacity of *different sources* in a network is usually not the same because of hardware limitations [13] of each computer or of topological limitation of the network. Further, the service capacity of each source is shown to vary with time [6, 7] due to the source workload or network congestion status. Both spatial heterogeneity in service capacity of different sources and temporal fluctuation of service capacity in time have significant impact on the average download time, as we will show later in the paper. The theoretical results obtained by the averaged value (F/c) is often too optimistic.

In this paper, we first characterize the relationship between the heterogeneity in service capacity and the average download time for each user, and show that the degree of diversity in service capacities has negative impact on the average download time. After we formally define the download time over a stochastic capacity process, we prove that the correlations in the capacity make the average download time much larger than the commonly accepted value F/c, where c is the average capacity of the connection. It is thus obvious that the average download time will be reduced if there exists a (possibly distributed) algorithm that can efficiently eliminate the negative impact of heterogeneity in service capacities over different paths and the negative impact of correlations in time of a given path.

In practice, most P2P applications try to reduce the download time by minimizing the risk of getting stuck in a 'bad' path (the connection with small service capacity) by using smaller file sizes and/or having them downloaded over different paths (e.g., parallel download). In other words, they try to reduce the download time by minimizing the *bytes* transferred from that source with small capacity. However, we show in this paper that this approach has almost no benefit at all in terms of explicitly reducing the average download time for each user in the network. We then propose a simple and distributed algorithm that can *effectively* remove the negative impact of both the correlations in the available capacity of a path and the heterogeneity in different sources and thus achieve the smallest possible download time. Through extensive simulations, we also verify that the proposed strategy consistently outperforms all the other schemes widely used in practice under almost all network configurations. In particular, the download time of our proposed scheme can be several times smaller than any other scheme when the network is heterogeneous (possibly correlated) and many downloading peers coexist with source peers, as mostly is the case in reality.

II. Characterizing the Download Time in a Network

We consider our network as a discrete-time system with each time slot of length Δ. For notational simplicity, throughout the paper, we will assume that the length of a time slot is normalized to one, i.e., $\Delta = 1$. Let $C(t)$ denote the time-varying service capacity (available end-to-end bandwidth) of a given path at time slot t ($t = 1, 2, \ldots$) over the duration of a download. Then, the download time T for a file of size F is defined as

$$T = \min\left\{ s > 0 \;\Bigg|\; \sum_{t=1}^{s} C(t) \geq F \right\}. \tag{1}$$

Note that T is a stopping time or the *first hitting time* of a process $C(t)$ to a fixed level F.

If $C(t)$, $t = 1, 2, \ldots$ are independent and identically distributed (i.i.d.), then by assuming an equality in (1), we obtain from Wald's equation [12] that

$$F = E\left\{ \sum_{t=1}^{T} C(t) \right\} = E\{C(t)\}E\{T\}. \tag{2}$$

The expected download time, measured in slots, then becomes $E\{T\} = F/E\{C(t)\}$. Note that (2) also holds if $C(t)$ is constant (over t). Thus, when the service capacity is *i.i.d.* over time or constant, there exists a direct relationship between the average service capacity and the average download time, as has typically been assumed in the literature.

II.A Impact of Heterogeneity in Service Capacity

We first consider the impact of heterogeneous service capacities of different paths. In order to decouple the effect of correlations from that of heterogeneity, in this section, we assume that Wald's equation holds true for *each path* (i.e., the service capacity of a given path is either constant or *i.i.d.* over time). But we allow the average capacities for different paths to be different. We will consider the impact of correlations in Section II.B.

Let N be the number of sources in the network (N different end-to-end paths) and $C_i(t)$ be the service capacity of path i at time slot t. We assume that $C_i(t)$ is either constant or *i.i.d.* over t such that (2) holds. Let $c_i = E\{C_i(t)\}$ be the average capacity of path i. Then, the average service capacity the network offers to a user becomes

$$A(\vec{c}) = \frac{1}{N}\sum_{i=1}^{N} c_i, \tag{3}$$

where $\vec{c} = (c_1, c_2, \ldots, c_N)$ and $A(\vec{c})$ is the arithmetic mean of the sequence $c_1, c_2, \ldots, c_N$. Thus, one may expect that the average download time, $E\{T\}$, of a file of size F would be

$$E\{T\} = \frac{F}{A(\vec{c})}. \tag{4}$$

As we mentioned earlier, however, the actual service capacity of each path remains hidden unless a network-wide probe is conducted. So the common strategy for a user is to randomly pick one source (path) and keep the connection to it until the download completes. If the user connects to path i (with service capacity $C_i(t)$), the average download time over that path becomes F/c_i from (2). Since the user can choose one of N paths with equal probability, the actual average download time in this case becomes

$$E\{T\} = \frac{1}{N}\sum_{i=1}^{N}\frac{F}{c_i} = \frac{F}{H(\vec{c})}, \tag{5}$$

where $H(\vec{c})$ is the harmonic mean of $c_1, c_2, \ldots, c_N$ defined by $H(\vec{c}) = [\frac{1}{N}\sum_{i=1}^{N}\frac{1}{c_i}]^{-1}$. Because $A(\vec{c}) \geq H(\vec{c})$ *, it follows that (5) $\geq$ (4). This implies that the actual average download time in a heterogeneous network is always larger than that given by 'the average capacity of the network' as in (4).

II.B First Hitting Time of a Correlated Stochastic Process

In this section we show that the expected first hitting time of a 'positively correlated process' is larger than that of an *i.i.d.* counterpart. Consider a fixed path between a downloading peer and its corresponding uploading peer for a file of size F. Let $C(t)$ be a stationary random process denoting the available capacity over that path at time slot t. We will assume that $C(t)$ is positively correlated over time. Then, as before, we can define the download time of a file (or the first hitting time of the process $C(t)$ to reach a level F) as T_{cor}, where the subscript '*cor*' means that $C(t)$ is a correlated stochastic process.

Suppose now that we are able to remove the correlations from $C(t)$. Let $C'(t)$ be the resulting process and T_{ind} be the stopping time for the process $C'(t)$ to reach level F, where the subscript '*ind*' now means that $C'(t)$ is independent over time. Then, again from Wald's equation, we have $E\{T_{ind}\} = F/E\{C'(t)\} = F/E\{C(t)\}$.

First, as introduced earlier, consider the case that $C(t)$ is 100% correlated over time, i.e., $C(t) = C$ for some random variable C for all t. Then, the download time T_{cor} becomes $T_{cor} = F/C$ assuming an equality in (1). Hence, from Jensen's inequality, we have

$$E\{T_{cor}\} = FE\left\{\frac{1}{C}\right\} \geq \frac{F}{E\{C\}} = E\{T_{ind}\},$$

i.e., the average first hitting time of an 100% correlated process is always larger than that of an *i.i.d.* counterpart.

*The arithmetic mean is always larger than or equal to the harmonic mean, where the equality holds when all c_i's are identical.

In order to characterize any degree of positive correlations in $C(n)$, we adopt the notion of 'association' [9, 12].

The set of associated processes comprises a large class of processes, and the most popular example is of the following type:

Theorem 4.3.13. in [9]: Let $\{X(t)\}$ be a stochastic process with static space $S = \mathbb{R}^d$ of the form

$$X(t+1) = \varphi(X(t), Z(t)), \quad \text{for} \quad t = 0, 1, \ldots \tag{6}$$

If the $\{Z(t)\}$ are mutually independent and independent of $X(0)$, then $\{X(t)\}$ is associated if $\varphi(x, z)$ is increasing in x. □

Stochastic processes of the form (6) constitute large portion of Markov processes. For example, any autoregressive type model with positive correlation coefficient can be written in the form of (6). Specifically, for an AR-1 sequence $X(t)$ defined by

$$X(t+1) = \rho X(t) + b\xi(t),$$

where $0 < \rho < 1$ and $\xi(t)$ $(t = 0, 1, \ldots)$ is a sequence of *i.i.d.* random variables and independent of $X(0)$, we can write $X(t+1) = \varphi(X(t), \xi(t))$ where $\varphi(x, \xi) = \rho x + b\xi$. Since φ is increasing in x, the process $\{X(t)\}$ is associated.

We below present our theorem. Due to space constraint, we refer to our technical report [3] for all the proofs.

Theorem 1 *Suppose that* $\{C(t), t \geq 1\}$ *is associated. Then, we have*

$$E\{T_{cor}\} \geq E\{T_{ind}\}. \tag{7}$$

□

Theorem 1 states that the average download time of a file over a path with correlated service capacity is always larger than that of an *i.i.d.* counterpart. In the subsequent section, we show the relationship between the degree of correlation of a process and the average first fitting time of that process, and illustrate how much $E\{T_{cor}\}$ can be larger than $E\{T_{ind}\}$.

II.C First Hitting Time and Degree of Correlation

Suppose that $C(t)$ is given by a stationary first-order autoregressive process (AR-1), i.e.,

$$C(t+1) = \rho \cdot C(t) + \epsilon(t) + \alpha. \tag{8}$$

Here, $\epsilon(n)$ is a sequence of *i.i.d.* random variables with zero mean, which represents a noise term of the process. Then, from the stationarity of the process, we get

$$E\{C(t)\} = \mu = \alpha/(1-\rho). \tag{9}$$

We vary the constant α such that the average capacity is always fixed to $E\{C(t)\} = \mu = 10$ under different ρ. Since the available bandwidth cannot be negative, we impose restriction on the range of $C(t)$ such that $C(t) \in [0, 20]$, while keeping the mean still the same. The file size is $F = 250$ and the noise term, $\epsilon(t)$, is chosen to be uniformly distributed over $[-1, 1]$, $[-5, 5]$, and $[-9, 9]$ to see how the noise term affects the average download time.

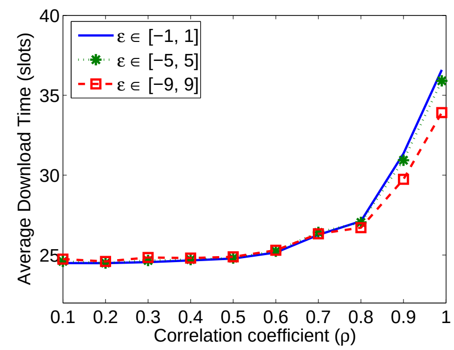


Figure 1: Relationship between the average download time and different degrees of correlation ρ under different noise term $\epsilon(t)$ in (8).

Figure 1 shows the relationship between the average download time and the degree of correlation of the process (8) for different ρ and $\epsilon(t)$. As the degree of correlation increases, the average download time increases. In particular, for a heavily correlated process, the average download time can be about 40% larger than for uncorrelated or lightly correlated process, regardless of different noise terms. In real data networks, the available capacity of a connection typically shows wild fluctuation; it becomes very low when congestion occurs, and it can reach up to the maximum link bandwidth when things go well. In addition, as technology advances, people are getting links of higher and higher speed, hence the range of available capacity fluctuation is also likely to increase. Therefore, it is very important to consider the effect of correlation in capacity over time when we calculate the average download time of a file transfer.

III. Minimizing Average Download Time Over Stochastic Channels

Given the characterization of the average download time in Section II, we can now analyze the performance of different strategies that help reduce the average download time for each user. In this section we will analyze the performance of (i) random chunk-based switching, and (ii) random time-based (periodic) switching. Both methods allow only one active link simultaneously. Although parallel downloading is another commonly used method to reduce file download time, as we will show in the simulation section, parallel downloading performs well only when there are very few users in the network and almost no heterogeneity in service capacity. In all other cases as in reality, the parallel downloading may even perform worse than single-link download. Hence we will focus on the performance analysis of single-link download strategies.

III.A Random Chunk-based Switching

In the random chunk-based switching, the file of interest is divided into many small chunks. A user downloads chunks sequentially one at a time. Whenever a user completes a chunk from its current source, the user randomly selects a new source and connects to it to retrieve a new chunk. In order to analyze its performance, we assume that Wald's equation holds as in (2) for each given path. A file of size F is divided into m chunks of equal size, and let t_j be the download time for chunk j. Then, the total download time, T_{chunk}, is $T_{chunk} = \sum_{j=1}^{m} t_j$. Since each chunk randomly chooses one of N sources (with equal probability), the expected download time will be

$$E\{T_{chunk}\} = \sum_{j=1}^{m} E\{t_j\} = \sum_{j=1}^{m} \frac{1}{N} \sum_{i=1}^{N} \frac{F/m}{c_i} = \frac{F}{H(\vec{c})}. \tag{10}$$

The result in (10) is identical to the download time given in (5) where a user downloads the entire file from an initially randomly chosen source. Downloading the entire file from one randomly chosen source or switching sources for different chunks makes no difference. Hence, the random chunk-based switching would not give us better performance in terms of per-user average download time in a multi-source network.

So far in this section, we have assumed that the service capacities in the network are heterogeneous, and Wald's equation holds true for a given path, i.e., the service capacity is either constant or *i.i.d.* over time. In this case, the random chunk-based switching does not give us any benefit. However, if the service capacity is correlated over time for each given path, switching sources may still help reduce the correlations. When there exist strong correlations, if we get stuck in a source with very low service capacity, it is likely that the service capacity from that source remains low for a while. Thus, instead of waiting until we finish downloading a fixed amount of data (chunk or file), we may want to get out of that bad source after some fixed amount of time. In other words, we can also think about random switching based on time. In the subsequent section, we will investigate the performance of this random switching based on time, and show that it outperforms all the previous schemes in the presence of heterogeneity of service capacities over space and temporal correlations of service capacity over each path.

III.B Random Periodic Switching

In this section, we propose a very simple, distributed algorithm and show that it effectively removes correlations in the capacity fluctuation and the heterogeneity in space, thus greatly reducing the average download time. Figure 2 shows a simplified model with multiple downloading users. We assume that each user can obtain a list of available sources through some search algorithm. As our algorithm will be implemented at each downloading peer in a distributed fashion, without loss of generality,

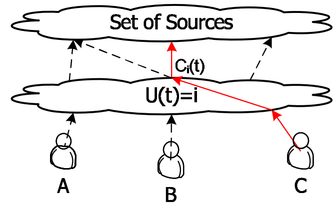


Figure 2: System model for file download operation in P2P networks

we only focus on a single downloader throughout this section.

In our model, there are N possible paths (source peers) for a fixed downloader. Let $C_i(t)$ ($t = 0, 1, 2, \ldots$ and $i = 1, 2, \ldots, N$) denote the available capacity during time slot t over path i (the connection between the fixed downloader and the i^{th} source peer). Let $U(t) \in \{1, 2, \ldots, N\}$ be a path selection function for the downloader. If $U(t) = i$, this indicates that the downloader selects path i and the available capacity it receives is $C_i(t)$ during the time slot t. For example, the solid arrow in Figure 2 represents a realization of the path selection function $U(t)$ at time t for a fixed downloader C. We assume that each $C_i(t)$ is stationary in t and $C_i(t)$ over different paths $i = 1, 2, \ldots, N$ are independent.† We however allow that they have different distributions, i.e., $E\{C_i(t)\} = c_i$ are different for different i (heterogeneity). For any given i, the available capacity $C_i(t)$ is correlated over time t. As before, when each connection has the same probability of being chosen, the average service capacity of the network is given by $A(\vec{c}) = \frac{1}{N}\sum_{i=1}^{N} c_i$.

In this setup, we can consider the following two schemes: *(i) permanent connection*, and *(ii) random periodic switching*. For the first case, the path selection function does not change in time t. When the searching phase is over and a list of available source peers is given, the downloader will choose one of them randomly with equal probability. In other words, $U(t) = U$ where U is a random variable uniformly distributed over $\{1, 2, \ldots, N\}$. For example, if the downloader chooses u ($u \in \{1, 2, \ldots, N\}$) at time 0, then it will stay with that path *permanently* ($U(t) = u$) until the download completes.

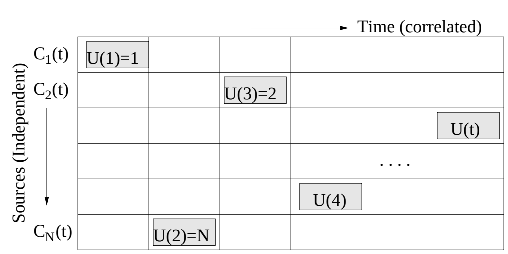


Figure 3: The operation of path selection function $U(t)$

†We note that different paths (overlay) may share the same link at the network core, but still, the bottleneck is typically at the end of network, e.g., access network type, or CPU workload, etc. Thus, the independence assumption here is reasonable.

For the periodic random switching, the downloader randomly chooses a path at each time slot, independently of everything else. In other words, the path selection function $U(t)$ forms an *i.i.d.* sequence of random variables, each of which is again uniformly distributed over $\{1, 2, \ldots, N\}$. Figure 3 illustrates the operation of the path selection function $U(t)$ for random periodic switching. In this figure, path 1 is selected at time 1, path N is selected at time 2, and so on.

Let us define an indicator function

$$I_u(t) = \begin{cases} 1, & \text{if } U(t) = u \\ 0, & \text{otherwise.} \end{cases}$$

Then, since $U(t)$ can take values only from $\{1, 2, \ldots, N\}$, the actual available capacity at time t can be written as

$$X(t) = C_{U(t)}(t) = \sum_{u=1}^{N} C_u(t) I_u(t)$$

for both the permanent connection and the random periodic switching strategies. Since each downloader chooses a path independently of the available capacity, $U(t)$ is also independent from $C_u(t)$, and so is $I_u(t)$. Note that, from $E\{I_u(t)\} = 1/N$ for any u, we have

$$E\{X(t)\} = \sum_{u=1}^{N} E\{C_u(t)\} E\{I_u(t)\} = \sum_{u=1}^{N} \frac{c_u}{N} = A(\vec{c}),$$

i.e., the average available capacity for the two path selection strategies are the same.

In order to analyze how the two different strategies affect the correlation in $X(t)$, we consider the correlation coefficient of $X(t)$ defined as

$$r(\tau) = \frac{\mathrm{Cov}\{X(t), X(t+\tau)\}}{\mathrm{Var}\{X(t)\}}.$$

Then, we have the following result.

Proposition 1 *Let $r_{per}(\tau)$ and $r_{ran}(\tau)$ denote the correlation coefficient of $X(t)$ under the permanent connection and the random periodic switching, respectively. Then, we have*

$$r_{ran}(\tau) \quad = \quad \frac{1}{N}\, r_{per}(\tau), \quad \forall\, t \geq 1.$$

□

From Proposition 1, we see that under the random periodic switching strategy, the correlation of $X(t)$ is N times smaller than that of permanent connection strategy. For example, when each downloader has about 10 available source peers ($N = 10$), the correlation coefficient of the newly obtained capacity process under our random periodic switching is no more than 0.1 regardless of the correlations present in the original capacity fluctuation. So, by using our random periodic switching, we can always make the capacity process *very lightly correlated*, or almost independent. From Figure 1, we see that the average download time for a lightly correlated process is very close to that given by Wald's equation. It is thus reasonable to assume that Wald's equation holds for the lightly correlated process $X(t)$ under our random periodic switching strategy. Specifically, if we define T_{ran} as the download time for a file of size F under the random periodic switching, we have

$$\begin{aligned} F &= E\left\{\sum_{n=1}^{T_{ran}} C_{U(t)}(t)\right\} = E\{T_{ran}\} E\{C_{U(t)}(t)\} \\ &= E\{T_{ran}\} E\left\{E\left\{C_{U(t)}(t) \mid U(t)\right\}\right\} \\ &= E\{T_{ran}\} \frac{1}{N} \sum_{u=1}^{N} E\{C_u(t)\} \\ &= E\{T_{ran}\} \frac{1}{N} \sum_{u=1}^{N} c_u = E\{T_{ran}\} A(\vec{c}). \end{aligned} \tag{11}$$

We then have the following comparison result between the permanent connection and periodic switching.

Proposition 2 *Suppose that the process $C_u(t)$ for each u is associated (i.e., it is correlated over time t). Let T_{per} and T_{ran} be the download time for the permanent connection and for the random periodic switching, respectively. Then, we have*

$$E\{T_{per}\} \geq E\{T_{ran}\}.$$

□

Proposition 2 shows that our random periodic switching strategy will always reduce the average download time compared to the permanent strategy and that the average download time under the random periodic switching is given by $F/(\vec{c})$ (see (4)). Note that this was made possible since the random periodic switching removes the negative impact of both the heterogeneity and the correlations. In addition, our algorithm is extremely simple and does not require any information about the system.

IV. Numerical Results

In this section we provide numerical results to support our analysis and compare the performance of the four schemes for file download under various network configurations. In any case, in our configuration, different paths have different average service capacities, and the service capacity of each path is correlated in time. We consider a single downloading peer as well as multiple downloading peers to allow competition among the downloading peers for limited service capacity of each source peer.

IV.A Single Downloader with Heterogeneous Path Characteristics

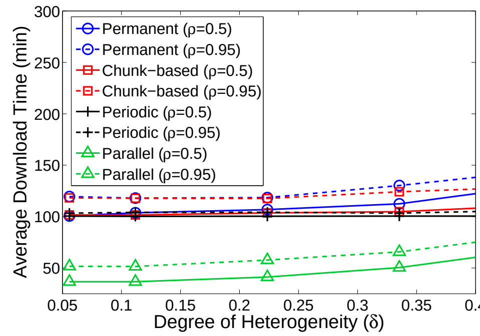
We first show the impact of both heterogeneity and correlations in service capacities on the average download time when there is a single user (downloader) in the network. There are $N = 4$ source peers in the network, each offering different average service capacities. Let c_i be the average service capacity of source i and $\vec{c} = (c_1, c_2, c_3, c_4)$. The average service capacity of the whole network is then $A(\vec{c}) = (c_1 + c_2 + c_3 + c_4)/4$. We change the heterogeneity in service capacity by changing each c_i, while keeping $A(\vec{c}) = 200$kbps the same. We measure the degree of heterogeneity in term of $\delta = \sqrt{\mathrm{Var}\{\vec{c}\}}/A(\vec{c})$, the normalized standard deviation. We set δ to range from 0.05 to 0.7.

To demonstrate the impact of correlation in each fixed path, we use a class of AR-1 random processes to model the stochastic fluctuation in the service capacity. The values of ρ and $\epsilon(t)$ in (8) represents the degree of correlation and the noise term of the process respectively. It is reasonable to assume that if the average service capacity is large, the service capacity is more likely to fluctuate over a wider range. In this regard, we assume that the amount of fluctuation in $C_i(t)$ is proportional to its mean value c_i. Specifically, for path i, we set $\epsilon_i(t)$ to be uniformly distributed over $[c_i - \theta_i, c_i + \theta_i]$ where θ_i is chosen such that $\sqrt{\mathrm{Var}\{C_i(t)\}}/E\{C_i(t)\}$ remains the same for all i.

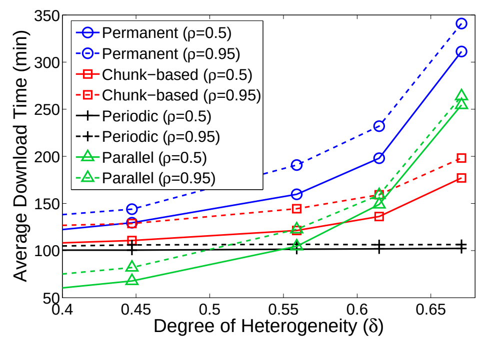
In our simulation, the network is modeled as a discrete time system where the length of each time slot (one period) is chosen to be 5 minutes. In reality, it is expected that within such a period, there is no major event that triggers dramatic fluctuation in the service capacity. There may be small short-term fluctuations, on the order of seconds, in the service capacity due to the nature of the network protocol, such as TCP congestion window changes, or OS interrupt handling, etc. These changes however do not impose serious impact on the service capacity. Thus, we are not interested in such small short-term variation, but are more interested in the fluctuation on a longer time scale caused by change in the number of connections at the source or change in network congestion status, which all usually last for longer time (say, minutes to hours).

We set the file size to 150MB, which is the typical size of some small video clips or multimedia files. As the average service capacity (of the network) is 200kbps, we set the chunk-size for chunk-based switching to be 7.5 MB (= 200kbps $\times$ 5 minutes) to allow fair comparison between periodic switching and chunk-based switching. Although many modern P2P systems make the size of a "chunk" to be around 256KB or so, it is unlikely for a peer to switch to different peers after each chunk. Rather, a downloading peer may receive several "chunks" consecutively from the same source peer. Hence the real data size downloaded from the same source is not just 256KB, but will be usually larger.

We consider four possible download strategies, permanent, periodic switching, chunk-based switching and parallel downloading. For permanent connection, the user initially choose one of four sources randomly and stay there until the download completes. For chunk-based switching, the peer switches to a new randomly selected source whenever a chunk is completed. Although we simulate the system as a discrete time system, the user is allowed to switch to a new source anytime within a time slot whenever it finishes the current chunk. For parallel download, the file is divided into 4 equal-sized pieces and the downloading peer connects to all 4 source and download each piece from each source peer simultaneously. Finally, for periodic switching, a user switches to a new randomly chosen source every 5 minute to further download the remaining parts of the file.



(a) Low degree of heterogeneity: $0 < \delta < 0.4$



(b) High degree of heterogeneity: $0.4 < \delta < 0.7$

Figure 4: Average download time vs. degree of heterogeneity under different download strategies and different degree of correlations.

Figures 4 (a)–(b) show the average download time vs. the degree of heterogeneity in the average service capacities (δ) when there is a single downloader in the network. Dashed lines are for strong correlations ($\rho = 0.95$) and solid lines represent the case of light correlations ($\rho = 0.5$). In Figure 4(a), when the degree of heterogeneity is small, all three single-link download strategies (permanent, chunk-based, periodic) under light correlations perform the same. This is well expected since the service capacities of all paths are almost *i.i.d.* over space and time, so switching doesn't make any difference and the average download time becomes $F/A(\vec{c}) = 150\text{MB}/200\text{kbps}$ $= 100$ minutes, as commonly used in practice. On the other hand, when there exists strong correlations in the service capacity, the download time is longer for all strategies except the periodic switching. For example, when $\delta = 0.1$, the correlation alone can cause more than 20% of increase in the average download time. Thus, when the network is more like homogeneous (i.e., small δ), the temporal correlation in the service capacity of each path becomes a major factor that renders the average download time longer. However, the average download time remains the same under the random periodic switching.

Figure 4 (a) also shows the performance of parallel downloading. Intuitively, parallel downloading should perform better than single link downloading because (i) it utilizes more than one link at the same time and (ii) if the connection is poor, parallel downloading reduces the amount of data getting through that bad path. Since there is only a single user, it utilizes all the service capacity the network can provide $(c_1 + c_2 + c_3 + c_4)$. In this case, the average download time should be $150\text{MB}/(c_1 + c_2 + c_3 + c_4) = 150\text{MB}/800\text{kbps} \approx 25$ minutes. We see from Figure 4(a) that parallel downloading can actually achieve the performance close to our expectation when the service capacities of different paths are close to *i.i.d.* Still, parallel downloading is prone to the negative effect of correlations.

As the degree of heterogeneity increases, the average download time sharply increases for all the schemes except the periodic switching. Figure 4 (b) shows this when δ is between 0.4 and 0.7. Permanent and chunk-based switching both suffer from the negative effect of heterogeneity. When both heterogeneity and correlation are high ($\delta = 0.65$ and $\rho = 0.95$), permanent connection takes about 350 minutes to complete the download. This time is about 250 minutes, or 4 hours more than using the periodic switching! Even the chunk-based switching yields the average download time almost as twice much as the periodic switching. Further, the performance of parallel downloading degrades very fast as the degree of heterogeneity increases. At some point, even for a single downloading peer in the network, the performance of parallel downloading becomes worse than the chunk-based switching (single-link download) and close to that of permanent connection. When there is a large degree of heterogeneity, it is more likely that one of the parallel connections is 'poor' with very small capacity. Thus, even though the size of chunk over each path is smaller than the whole file (hence reducing the risk of staying with the bad path for too long), this is still not as good as the idea of averaging capacities all the time, as used in the periodic switching. We note that temporal correlations still negatively affect in all these three schemes. However, it should be pointed out that the random periodic switching performs the same *regardless of heterogeneity and correlations*, and in fact it outperforms all the other schemes when the network is heterogeneous with a wide range of service capacities as in the current network.

IV.B Multiple Downloaders with Competition

As seen earlier, when there is only one downloading peer in the network, the average download time can be reduced by randomly switching connections among paths with different average capacities. However, there are always multiple users in a network and competition among peers is obvious. In this section, we consider the performance of different download strategies under a multi-user environment. In the multi-user scenario, we set the number of source peers to $N = 100$. The sources are divided into 4 groups and each source within the same group will have the same average service capacity. We set the correlation coefficient of each source to 0.5, which indicates light temporal correlations in the service capacity. The distribution of service capacity of each group is chosen to make $\delta = 0.6$. In our simulation, if m users connect to a source offering service capacity $C(t)$, then each of those m users receives service capacity of $C(t)/m$. The effect of dividing capacity among users gives us an idea of how different strategies will perform when users compete for limited resources in the network. To represent the level of competition, we use the *source-user* ratio, i.e. the ratio between the number of source peers to the number of users (downloading peers).

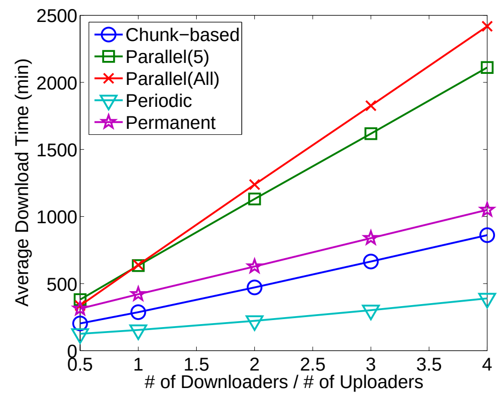


Figure 5: Performance of different download strategies under if different levels of competition.

Figure 5 shows the average download time under different strategies as the number of downloaders increases (different levels of competition). Because the total amount of service capacity in the network is fixed, users have to share this fixed amount of capacity and thus the average download time will increase as the number of downloaders increases for all possible strategies. In Figure 5, the periodic switching always yields the smallest average download time than all the other strategies regardless of the level of competition among downloading peers. Even under heavy competition, when the number of users is 4 times of the number of sources, periodic switching can reduce at least 50% of the average download time compared to other strategies.

It is interesting to see that under heavy competition, parallel downloading is even worse than the single link download. As discussed earlier, in parallel downloading, the download time is determined by the chunk that finishes the last. In other words, the download is complete when the chunk from the source offering the 'worst' service capacity is done. Under heavy competition, parallel downloading actually makes more users to share the 'worst' source, thus increasing the download time further. Thus, our study shows that, contrary to the common belief, parallel downloading is far from being optimal in terms of reducing the download time. Its performance is very much dependent upon the heterogeneity in service capacities in the network and upon the degree of competition in the network. In a network consisting of many heterogeneous sources (as in current P2P networks), we are better off keeping fewer connections so that not every user gets stuck in the bad sources. Overall, from Figures 4, it is evident that the random periodic switching generally performs the best. Although it may not compete with parallel downloading in the case where all sources offer the same service capacity with very light competition, those conditions are not practical in most cases. For all other situations with realistic heterogeneity, competition among peers, as well as with temporal correlations in each path, the random periodic switching gives the most stable and optimal performance in the sense that its download time is minimal, robust with respect to network configurations, and consistent for different users in a random environment.

References

[1] M. Adler, R. Kumar, K. Ross, D. Rubenstein, D. Turner, and D. D. Yao. Optimal peer selection in a free-market peer-resource economy. In *Proceedings of Workshop on Economics of Peer-to-Peer Systems (P2PEcon)*, Cambridge, MA, Jun. 2004.

[2] M. Adler, R. Kumar, K. Ross, D. Rubenstein, D. Turner, and D. D. Yao. Optimal peer selection for p2p downloading and streaming. In *Proceedings of IEEE Infocom*, Miami, FL, Mar. 2005.

[3] Y. Chiu and D. Y. Eun. Minimizing File Download Time over Stochastic Channels in Peer-to-Peer Networks. Technical report, North Carolina State University, Raleigh, NC, Dec. 2005. available at "http://www4.ncsu.edu/∼dyeun/pub/techrep-ciss06-chiu.pdf".

[4] G. Bianchi F. Lo Piccolo, G. Neglia. The effect of heterogeneous link capacities in bittorrent-like file sharing system. In *IEEE International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P)*, Oct. 2004.

[5] K. P. Gummadi, R. J. Dunn, and S. Saroiu. Measurement, modeling, and analysis of a peer-to-peer file sharing workload. In *Proceedings of ACM Symposium on Operating Systems Principles (SOSP)*, 2003.

[6] N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE Journal on Selected Areas in Communications*, 21(6), Mar. 2003.

[7] M. Jain and C. Dovrolis. End-to-end estimation of the available bandwidth variation range. In *Proceedings of ACM Sigmetrics*, Jun. 2005.

[8] S. Koo, C. Rosenberg, and D. Xu. Analysis of parallel downloading for large file distribution. In *Proceedings of IEEE International Workshop on Future Trends in Distributed Computing Systems (FTDCS)*, May 2003.

[9] A. Müller and D. Stoyan. *Comparison Methods for Stochastic Models and Risks.* John Wiley & Son, New York, NY, 2002.

[10] D. Qiu and R. Srikant. Modelling and performance analysis of bittorrent-like peer-to-peer networks. In *Proceedings of ACM Sigcomm*, Aug. 2004.

[11] K. K. Ramachandran and B. Sikdar. An analytic framework for modelling peer to peer networks. In *Proceedings of IEEE Infocom*, Mar. 2005.

[12] S. M. Ross. *Stochastic Processes.* John Wiley & Son, New York, second edition, 1996.

[13] S. Saroiu, K. P. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceegins of ACM Multimedia Computing and Networking (MMCN)*, 2002.

[14] X. Yang and G. de Veciana. Service capacity of peer to peer networks. In *Proceedings of IEEE Infocom*, Mar. 2004.