Analysis of Peer-to-Peer Networks and a Study on File Download Minimization

Kolan Helini, P. Prashanthi and G. Radha Devi
Samskruti College of Engineering and Technology, India

Abstract– **The peer-to-peer file-sharing applications are becoming increasingly popular and account for more than 70% of the Internet's bandwidth usage. Measurement studies show that a typical download of a file can take from minutes up to several hours depending on the level of network congestion or the service capacity fluctuation. In this paper, we consider two major factors that have significant impact on average download time, namely, the spatial heterogeneity of service capacities in different source peers and the temporal fluctuation in service capacity of a single source peer. We point out that the common approach of analyzing the average download time based on average service capacity is fundamentally flawed. We rigorously prove that both spatial heterogeneity and temporal correlations in service capacity increase the average download time in P2P networks and then analyze a simple, distributed algorithm to effectively remove these negative factors, thus minimizing the average download time. We show through analysis and simulations that it outperforms most of other algorithms currently used in practice under various network configurations.**

Keywords– Peer-to-Peer, Stochastic Stability and Stochastic Coupling

I. INTRODUCTION

Before we start the topic, let us understand what is meant by Stochastic and Spatial heterogeneity. Stochastic means a process of which the outcome appears to be unpredictable**.** Spatial heterogeneity is a property generally ascribed to a landscape or to a population. It refers to the uneven distribution of various concentrations of each species within an area. A population showing spatial heterogeneity is one where various concentrations of individuals of this species are unevenly distributed across an area; nearly synonymous with "patchily distributed. So let us consider that the species here are computers.

Spatial analysis confronts many fundamental issues in the definition of its objects of study, in the construction of the analytic operations to be used, in the use of computers for analysis, in the limitations and particularities of the analyses which are known, and in the presentation of analytic results. Many of these issues are active subjects of modern research. In probability theory, a stochastic process or sometimes random process (widely used), is the counterpart to a deterministic process (or deterministic system). Instead of dealing with only one possible way the process might develop over time (as in the case, for example, of solutions of an ordinary differential equation), in a stochastic or random process there is some indeterminacy described by probability distributions. This means that even if the initial condition (or starting point) is known, there are many possibilities the process might go [1] to, but some paths may be more probable and others less so.

In the simplest possible case a stochastic process amounts to a sequence of random variables known as a time series. Another basic type of a stochastic process is a random field, whose domain is a region of space, in other words, a random function whose arguments are drawn from a range of continuously changing values. One approach to stochastic processes treats them as functions of one or several deterministic arguments (inputs, in most cases regarded as time) whose values (outputs) are random variables: non-deterministic (single) quantities which have certain probability distributions. Random variables corresponding to various times (or points, in the case of random fields) may be completely different. The main requirement is that these different random quantities all have the same type. Type refers to the co-domain of the function. Although [2] the random values of a stochastic process at different times may be independent random variables, in most commonly considered situations they exhibit complicated statistical correlations.

Familiar examples of processes modeled as stochastic time series include stock market and exchange rate fluctuations, signals such as speech, audio and video, medical data such as a patient's EKG, EEG, blood pressure or temperature, and random movement such as Brownian motion or random walks. Examples of random fields include static images, random terrain (landscapes), or composition variations of a heterogeneous material.

II. SURVEY ON PEER-TO-PEER COMMUNICATIONS

Peer-to-peer (P2P) communication in the Internet is provided through the sharing of widely distributed resources typically involving end users' computers acting as both clients and servers. In an unstructured peer-to-peer network, such as *BitTorrent*, a file is divided into many pieces. Seeds, which hold all pieces, distribute pieces to peers. New peers continually arrive into the network; they simultaneously download pieces from a seed or other peers and upload pieces to other peers. Peers exit the system after they collect all pieces. Determining whether a given P2P network is stable can be difficult. Roughly speaking, the aggregate transfer capacity scales up in proportion to the number of peers in the network, but it has to be in the right places.

Many P2P systems have performed well in practice, and they incorporate a variety of mechanisms to help achieve stability. A broad problem, which we address in part, is to

provide a better understanding of which mechanisms are the most effective under various network settings. These mechanisms include:

- *Rarest first piece selection policies,* such as the one implemented in BitTorrent, whereby peers determine which pieces are rarest among their neighbors and preferentially download such pieces
- *Tit-for-tat participation constraints,* such as the one implemented in BitTorrent, whereby peers are choked off from receiving pieces from other peers unless they upload pieces to those same peers. This mechanism provides an important incentive for peers to participate in uploading pieces, but it may also be beneficial in balancing the distribution of pieces
- *Peers dwelling in the network after completing download,* to provide extra upload capacity
- *Network coding* whereby data pieces are combined to form coded pieces, giving peers numerous ways to collect enough information to recover the original data file

In this paper, we [3] investigate the impact of the interaction and competition among peers on downloading performance under stochastic, heterogeneous, unstructured P2P settings, thereby greatly extending the existing results on stochastic P2P networks made only under a single downloading peer in the network. To analyze the average download time in a P2P network with multiple competing downloading peers, we first introduce the notion of system utilization tailored to a P2P network. We investigate the relationship between the average download time, system utilization and the level of competition among downloading peers in a stochastic P2P network. We then derive an achievable lower bound on the average download time and propose algorithms to give the peers the minimum average download time. Our result can much improve the download performance compared to earlier results in the literature.

A. Related Work on Peer-to-Peer Technology

Peer-to-peer (P2P) technologies, such as Bit Torrent have been widely used for file transfer over the Internet. In those applications, file download time is one of the most important performance metrics. Theoretically, a P2P network can make its users download faster compared to a traditional client-server network because a P2P network is inherently scalable. Each node in a P2P network can act both as a server and a client *simultaneously*. As a result, the aggregate system service capacity increases with the number of downloading nodes (demand) in a P2P network. It is widely believed that only the physical access bandwidth of each downloading peer can limit the download performance.

However, measurement studies show rather counterintuitive results. It is shown in that downloading a 100MB file from a Gnutella or Kaaza network generally takes hours often up to a week. Considering the fact that earlier P2P studies predicted that the download performance is only limited by each peer's physical access bandwidth and most people nowadays have broadband access, the typical download time for a 100MB file should be less than an hour if the prediction is correct. The gap between prediction and measured performance motivated us to investigate more about the file download process in a P2P system. Most people believe that the existence of free-riders is the main reason that degrades the performance of a P2P network. Accordingly, many incentive algorithms have been developed so as to encourage peers to contribute to the network.

However, even if all peers in the network are altruistic, we are still far away from enjoying the performance predicted by Results in suggest that *both* the stochastic temporal fluctuation and the heterogeneity in service capacity of each peer can make the average download time significantly longer than expected, even when all peers in the network are willing to share. Although some other earlier studies have also noticed the impact of stochastic fluctuation and the heterogeneity in service capacity of each peer, those studies often have more limited viewpoints. For example, the researchers some times [4] consider only the stochastic fluctuation in service capacity of each peer but they do not consider the network heterogeneity. On the other hand, they try to develop an optimal peer selection method that exploits the heterogeneity of the network but do not consider the temporal fluctuation in service capacity of each peer. More important, all the results in have been established under the assumption that *there is only one downloading peer in the network*. This is critical, since in a real P2P network there will be multiple peers uploading and downloading at the same time and a peer's service capacity will be shared by its competing peers. In other words, the downloading peers will have to compete for the limited resource each single source peer can offer. In this setting, the download performance is determined not only by the stochastic fluctuation and heterogeneity in service capacity that each peer offers; how each peer makes its peer selection choice under such a competitive environment is also very important.

B. Challenges in Peer-to-Peer Networks

Some of the major challenges facing a P2P network in the real world include peer selection, and data search and routing. Due to the distributed nature of the P2P network, searching and locating data of interest in the network has been an important issue in the literature. In reality, data searching time only contributes a very small portion of a download session while the most delay is caused by actually transferring the file from source peers. Thus, if we want to minimize the download time for each user, reducing the actual file transfer time would make more noticeable difference. Most recent studies, however, have focused on reducing the total download duration, i.e. the time required for *all users* to finish their download. This total download time is a system-wide performance metric.

On the other hand, there are very few results in analyzing the performance of *each individual user*. As the measurement study shows, the per-user performance in a P2P network may be even worse than that of centralized network architecture. Those results suggest that there is much room for improvement in the P2P system in terms of per-user performance, i.e. the file download time of each user. However, there have been very few results in minimizing the download time for *each user* in a P2P network. In recent

work, the problem of minimizing the download time is formulated as an optimization problem by maximizing the aggregated service capacity over multiple simultaneous active links (parallel connections) under some global constraints. There are two major issues in this approach. One is that global information of the peers in the network is required, which is not practical in real world. The other is that the analysis is *based on the averaged quantities*, e.g., average capacities of all possible source peers in the network. The approach of using the average service capacity to analyze the average download time has been a common practice in the literature

III. LIST OF APPROACHES ARE AVERAGE SERVICE CAPACITY

We here illustrate limitations of the approach based on averaged quantities in a P2P network by considering the following examples. Suppose that a downloading peer wants to download a file of size F from N possible source peers. Let ci be the average end-to-end available capacity between the downloading peer and the ith source peer (i = 1, 2, . . . ,N). Notice that the actual value of ci is unknown before the downloading peer actually connects to the source peer i.

The average service capacity of the network, $\bar{C}$, is given by $\bar{C} = \sum_{i=1}^{N} c_i/N$

Intuitively, the average download time, T, for a file of size F would be

$$T = F/\bar{C}.$$

In reality, however, (1) is far different from the true average download time for each user in the network. The two main reasons to cause the difference are (i) the spatial heterogeneity in the available service capacities of different end-to-end paths and (ii) the temporal correlations in the service capacity of a given source peer. We first consider the impact of heterogeneity. Suppose that there are two source peers with service capacities of c1 = 100kbps and c2 = 150kbps, respectively, and there is only one [5] downloading peer in the network. Because the downloading peer does not know the service capacity of each source peer 1 prior to its connection, the best choice that the downloading peer can make to minimize the risk is to choose the source peers with equal probability. In such a setting, the average capacity that the downloading peer expects from the network is (100+150)/2 = 125kbps. If the file size F is 1MB, we predict that the average download time is 64 seconds. However, the actual average download time is 1/2(1MB/100kbps)+1/2(1MB/150Mbps) = 66.7 seconds! Hence, we see that the spatial heterogeneity actually makes the average download time longer.

Suppose now that the average service capacity can be known before the downloading peer makes the connection. Then, an obvious solution to the problem of minimizing the average download time is to find the peer with the maximum average capacity, i.e., to choose peer j with the average capacity cj (cj ≥ ci for all i), as the average download time Ti over source peer i would be given by F/ci. We assume that each peer can find the service capacity of its source peers via packet-level measurements or short-term in-band probing. Consider again the previous two-source peer example with c1 = 100kbps and c2 = 150kbps. As we want to minimize the download time, an obvious choice would be to choose source peer 2 as its average capacity is higher. Now, let us assume that the service capacity of source peer 2 is not a constant, but is given by a stochastic process C2(t) taking values 50 or 250kbps with equal probability, thus giving E{C2(t)} = c2 = 150kbps. If the process C2(t) is strongly correlated over time such that the service capacity for a file F is likely to be the same throughout the session duration, it takes on average (1MB/50kbps + 1MB/250kps)/2 = 96 seconds, while it takes only 80 seconds to download the file from source peer 1.

In other words, it may take longer to complete the download when we simply choose the source peer with the maximum average capacity! It is thus evident that the impact of correlations (second-order statistics) or higher-order statistics associated with the capacity fluctuation in time will need to be taken into account, even for finding a source peer with minimum *average* download time. In practice, most P2P applications try to reduce the download time by minimizing the risk of getting stuck with a 'bad' source peer (the connection with small service capacity) by using smaller file sizes and/or having them downloaded over different source peers (e.g., parallel download).2 In other words, they try to reduce the download time by minimizing the *bytes* transferred from the source peer with small capacity.

However, we show in this paper that this approach cannot effectively remove the negative impact of both the correlations in the available capacity of a source peer and the heterogeneity in different source peers. This approach may help to reduce average download time in some cases but not always. Rather, a simple and distributed algorithm that limits the amount of *time* each peer spends on a bad source peer, can minimize the average download time for each user almost in all cases. Through extensive simulations, we also verify that the simple download strategy outperforms all other schemes widely used in practice under various network configurations. In particular, both the average download time and the variation in download time of our scheme are smaller than any other scheme when the network is heterogeneous (possibly correlated) and many downloading peers coexist with source peers, as is the case in reality.

A. Quality of Heterogeneous Service Capacity

In a P2P network, just like any other network, the service capacities from different source peers are different. There are many reasons for this heterogeneity. On each peer side, physical connection speeds at different peers vary over a wide range (e.g., DSL, Cable, T1, etc). Also, it is reasonable to assume that most peers in a typical P2P network are just personal computers, whose processing powers are also widely different. The limitation in the processing power can limit how fast a peer can service others and hence limits the service capacity.

On the network side, peers are geographically located over a large area and each logical connection consists of multiple hops. The distance between two peers and the number of hops surely affect its round-trip-time (RTT), which in turns affects the throughput due to the TCP congestion

control. Moreover, in a typical P2P network, this information is usually 'hidden' when a user simply gets a list of available source peers that have contents the user is looking for.

Note that the aforementioned factors do not change over the timescale of any typical P2P session (days or a week). Hence, we assume that those factors mainly determine the long-term averages of the service capacity over a given source peer.

Correlations in Service Capacity: While the long-term average of the service capacity is mainly governed by topological parameters, the actual service capacity during a typical session is never constant, but always fluctuates over time. There are many factors causing this fluctuation. First, the number of connection a source peer allows is changing over time, which creates a fluctuation in the service capacity for *each user*. Second, some user applications running on a source peer (usually a PC), such as online games, may throttle the CPU and impact the amount of capacity it can offer. Third, temporary congestion at any link in the network can also reduce the service capacity of all users utilizing that link.

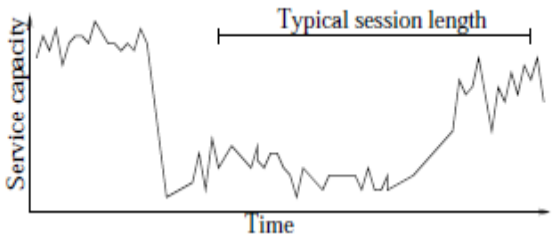

Fig. 1. Typical variation in End-to-End available bandwidth

Fig. 1 shows a typical available end-to-end capacity fluctuation. The time scale for the figure in the measurement study is on the order of minutes. We know that a typical file download session can last from minutes to hours for a file size of several megabytes. This implies that the service capacity over the timescale of one download session is stochastic and correlated. In Fig. 1, the short-term variations in the capacity are mainly due to the window size fluctuation in TCP, while the long-term variations are due to network congestion, changes in workload or the number of connecting users at the source peer, etc. The long-term fluctuation typically lasts over a longer time scale, say, few minutes up to several hours. As illustrated in the introduction, both the heterogeneity over different source peers and the correlations of the capacity in a given source peer have significant impact on the average download time. To the best of our knowledge, however, there has been no result available in the literature addressing these issues. All the existing studies have simply assumed that the service capacity is given by a constant (its average value) for the duration of a download. Consequently, the download time of a file of size F is simply given by F/c, where c is the average service capacity. As will be seen later on, however, this is true only when the service capacity is constant or i.i.d. over time, neither of them is true in reality.

B. Issues in Peer-to-Peer Networks

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

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

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, . . . are independent and identically distributed (i.i.d.), then by assuming an equality in (2), we obtain from Wald's equation that

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

The expected download time, measured in slots, then becomes E{T} = F/E{C(t)}. Note that (3) 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 section II.

IV. AVERAGE DOWNLOAD TIME FOR STOCHASTIC PEER-TO-PEER NETWORKS

Intuitively, if a downloader relies on a single source peer for its entire download, it risks making an unlucky choice of a slow source resulting in a long download. Since the service capacity of each source peer is different and fluctuates over time, utilizing different source peers either simultaneously (parallel downloading) or sequentially within one download session would be a good idea to diversify the risk. Parallel downloading improves the performance by reducing the file size over the 'worst' source peer and also may increase the service capacity one receives from the network by utilizing 'unused' capacities of other source peers. If a downloader utilizes one source peer at a time, switching around seems to be a good strategy to avoid the 'bad' source peer. Now, the question is, "What is the criterion for switching, i.e., is it chunk-based or time-based?" In this section we will analyze the performance of (i) parallel downloading, (ii) random chunk-based switching, and (iii) random time-based (periodic) switching.

Different strategies have different impact on the average download time of each peer, which may result in different system dynamics as well, e.g., how fast a downloader can start to contribute (become a source peer) or how fast a peer leaves the system after finishing download. If there is no peer leaving the system and all peers are willing to share after they complete their download (either the entire file or a chunk), the aggregate service capacity in the system keeps increasing as time goes on because the number of source peers continuously grows. In this case, the dynamics in the increase

of aggregate service capacity becomes the dominant factor in the average download time for each peer.

On the other hand, if no peer is willing to share after download, the aggregate capacity will then eventually drop to zero, thus throttling all the performance metrics. In reality, however, the P2P network will reach a steady-state at some point in which the peer arrivals and departures are balanced and the aggregate service capacity remains around some constant with little variation. This suggests that the number of source peers in the system will also be around some constant with little fluctuation in the steady-state. In this paper, we are mostly 6 interested in the impact of stochastic variations of capacities on the average download time of each peer in the steady-state, rather than in the impact of sources-downloaders dynamics in the transient period, which is beyond the scope of this paper.

Before we start our analysis, we have the following assumptions:

(i) The service capacity of a source is constant within one time slot.

(ii) Each downloader selects its source independently.

(iii) Each downloader makes blind choice, i.e. the sources are randomly chosen uniformly over all available sources.

Assumption (i) is reasonable since it is expected that there is no major event that triggers dramatic fluctuation in the service capacity within a short period of time. 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 variations, but are more interested in the fluctuation on a longer time scale caused by change in the number of connections at a source peer or change in network congestion status, which all usually last for longer time (say, minutes to hours).

We have the assumption (ii) because it is impractical for any downloader to know how other downloaders choose their source peers in the network. Hence, the downloader cannot not make its source selection decision based on other downloaders' decision.

Assumption (iii) is based on the fact that the downloader does not know the service capacity of each source peer a priori. Although some protocols require peers to broadcast information about its physical connection speed, it is hard to tell the "true" instant service capacity of each source peer due to many factors such as competition among other peers, changing workload of the source peer, or the network congestion status. Therefore, a simple way to select a source peer is just to make blind choice.

A. While Downloading What We Face of Parallel Downloading

Parallel downloading is one of the most noticeable way to reduce the download time. If the file F is divided into k chunks of equal size, and k simultaneous connections are used, the capacity for this download session becomes c1+ c2+. . .+ck, where ci is the service capacity of ith connection. Intuitively, this parallel downloading seems to be optimal in all cases. But, it is worth noting that the download time for parallel downloading is given by max{t1, t2, . . . , tk} rather than F/(c1 + c2 + . . . + ck), where ti is the download time of a chunk over ith connection. This is because the chunk that takes the longest time to complete determines the entire download session.

To illustrate that parallel downloading is better than single download, we consider the following simple example. Assume that there are only two source peers in the network, and c1, c2 are the service capacities of the two source peers. Without loss of generality, we assume that c1 ≤ c2. If parallel downloading is used for downloading a file of size F from the network, the

download time T_p is given by

$$T_p = \max\left\{\frac{F}{2c_1}, \frac{F}{2c_2}\right\} = \frac{F}{2c_1}.$$

For the case of single download, the average download time $\mathbb{E}\{T_s\}$ is

$$\mathbb{E}\{T_s\} = \frac{1}{2}\left(\frac{F}{c_1} + \frac{F}{c_2}\right) > \mathbb{E}\{T_p\} = T_p.$$

Now, given that parallel download is better than single download, one may ask whether it is as good as the predicted value in (1). To answer this, let's recall the two-source peers example. From (1), the predicted download time is

$$\mathbb{E}\{T\} = \frac{F}{A(\vec{c})} = \frac{2F}{c_1 + c_2}.$$

An easy calculation shows E{T} < E{Tp} if c2 > 3c1. Thus, even in the network with one user, parallel downloading may not reduce the download time to the predicted value in all cases. Instead, the performance of parallel download depends upon the distribution of the underlying service capacities and could be much worse than the ideal case, F/A(~c). Indeed, it is show that if we can make the chunk-size proportional to the service capacity of each source peer, parallel downloading can yield the optimal download time. But such scheme requires global information of the network. One of our goals is to find a simple and distributed algorithm with *no global information* such that the value in F/A(~c), can be achieved under almost all network settings. We have already seen that parallel downloading may not achieve F/A(~c) even when there is only one user in the network. Further, it is shown that in a multi-user network, maintaining just a few parallel connections, say, 4 to 6, is better than having parallel connections to all possible source peers. Hence, if there is an algorithm that can increase the performance of *each individual* connection among such a few parallel connection, then each individual user may achieve the download time predicted or even better.

B. Chunk-based Switching in Peer-to-Peer Networks in Random

In the random chunk-based switching scheme, the file of interest is divided into many small chunks just as in the parallel download scheme. A user downloads chunks sequentially one at a time. Whenever a user completes a chunk from its current source peer, the user randomly selects a new source peer and connects to it to retrieve a new chunk. In this way, if the downloader is currently stuck with a bad source peer, it will stay there for only the amount of time required for finishing one chunk. The download time for one chunk is independent of that of the previous chunk. Intuitively, switching source peers based on chunk can reduce the correlation in service capacity between chunks and hence reduce the average download time.

However, there is another factor that has negative impact on the average download time, the spatial heterogeneity. First, suppose that there is no temporal correlation in service capacity and Wald's equation holds for each source peer. A file of size F is divided into m chunks of equal size, and let tj be the download time for chunk j. Then, the total downloads time,

T_{chunk}, is $T_{chunk} = \sum_{j=1}^{m} t_j$. Since each chunk randomly chooses one of N source peers (with equal probability), the expected download time will be

$$\mathbb{E}\{T_{chunk}\} = \sum_{j=1}^{m} \frac{1}{N} \sum_{i=1}^{N} \frac{F/m}{c_i} = \frac{F}{H(\vec{c})}. \qquad (18)$$

The result in the formula above is identical to the download time given in where a user downloads the entire file from an initially randomly chosen source peer. In other words, the chunk-based switching is still subject to the 'curse' of spatial heterogeneity. While there is no benefit of the chunk-based switching from the average download time point of view, it turns out that this scheme still helps reduce the variance of the download time under a relatively smaller number of users by diversifying the risk with smaller chunks.

In the chunk-based switching, if we get stuck in a source peer with very low service capacity, downloading a fix amount of bytes from that source peer may still take a long time. We could avoid this long wait by making the size of each chunk very small, but this then would cause too much overhead associated with switching to many source peers and integrating those many chunks into a single file. Therefore, 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 peer after some fixed amount of time. In other words, we randomly switch 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 of each source peer.

C. Periodic Switching in Random Peer-to-Peer Networks

In this section, we analyze 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. As the algorithm will be implemented at each downloading peer in a distributed fashion, without loss of generality, we only focus on a single downloader throughout this section. In our model, there are N possible source peers for a fixed downloader. Let Ci(t) (t = 0, 1, 2, . . . and i = 1, 2, . . . ,N) denote the available capacity during time slot t of source peer i. Let U(t) □ {1, 2, . . . ,N} be a source selection function for the downloader. If U(t) = i, this indicates that the downloader selects path i and the available capacity it receives is Ci(t) during the time slot t. We assume that each Ci(t) is stationary in t and Ci(t) of different source peers i = 1, 2, . . . ,N are independent.4 We however allow that they have different distributions, i.e., E{Ci(t)} = ci are different for different I (heterogeneity). For any given i, the available capacity Ci(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 source 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, . . . ,N}. For example, if the downloader chooses u (u ∈ {1, 2, . . . ,N}) at time 0, then it will stay with that source peer *permanently* (U(t) = u) until the download completes.

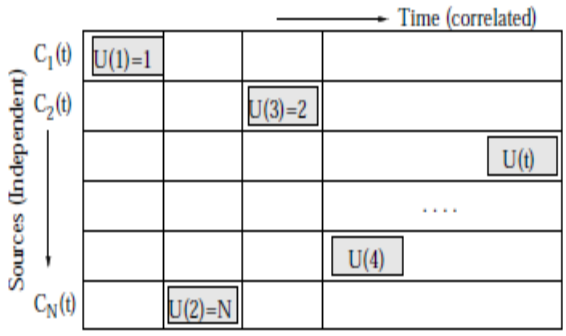

Fig. 2. Operation of Random Periodic Function

The Fig. 2 above shows the operation of source selection function U(t) for random periodic matching. For the random periodic switching, the downloader randomly chooses a source peer at each time slot, independently of everything else. In other words, the source selection function U(t) forms an i.i.d. sequence of random variables, each of which is again uniformly distributed over {1, 2, . . . ,N}. Fig. 3 illustrates the operation of the source selection function U(t) for random periodic switching. In this Fig. 3, source 1 is selected at time 1, source 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, . . . ,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 source peer independently of the available capacity, U(t) is also independent from Cu(t), and so is Iu(t). Note that, from

$$\mathbb{E}\{I_u(t)\} = 1/N \text{ for any } u, \text{ we have}$$

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

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

i.e., the average available capacity for the two source 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 tof X(t) defined as

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

V. COMPARATIVE STUDY

So far, we have analyzed the performance of three different schemes that utilize the spatial diversity of the network to improve per-user performance in terms of the average download time. We have considered (i) parallel downloading, (ii) random chunk-based switching, and (iii) random periodic switching. The parallel downloading may perform well if the capacity of each possible source peer is known so as to allocate larger chunks to faster connections and smaller chunks to slower connections. But this method is not practical as one cannot know a priori the service capacity of all source peers. In addition, the service capacity is stochastically fluctuating all the time, and our analysis show that the performance of parallel downloading depends much upon the heterogeneity of the service capacities in different source peers if the chunks are equal in size.

Many P2P applications nowadays use chunk-based file transfer with equal chunk size. As mentioned earlier, the benefit of chunk-based switching is to speed up the conversion from downloading peers to uploading peers and thus indirectly affect the average download time. But, in terms of reducing the average download time directly, it does not help much. Random chunk-based switching may reduce the correlations in the service capacity, but it still cannot eliminate the effect of spatial heterogeneity in different source peers. In current practice, the chunk based transfer and the parallel download are often combined. Taking BitTorrent and Overnet for examples, a file is first divided into 256KB and 9.5MB chunks of equal size, respectively, and then different chunks are downloaded from different source peers simultaneously.

However, we separate the analysis of the two strategies to show how each is different in combating spatial heterogeneity and temporal correlations. Please note that we are not trying to compare the performance of parallel downloading with chunk based transfer since they can be easily combined to yield better performance. Rather, we are comparing the performance of the two ategies with our random periodic scheme. The idea of time-based switching scheme is in fact not new. Such strategy has been implemented in BitTorrent but with some other purpose in mind. In BitTorrent application, by using its optimistic choking/unchoking algorithm, a peer changes one of its servicing neighbors with the lowest upload capacity every 10 seconds in hope to find some peers offering higher service capacity. However, the idea of switching source peer periodically in the BitTorrent's optimistic choking/unchoking algorithm is to discover new potential sources rather than to explicitly remove the negative impact of temporal correlations and spatial heterogeneity in service capacity. To the best of our knowledge, we are the first to point out that the random periodic switching gives us the average download time of F/A(~c), while all the other schemes considered so far yield larger average download time.

Our study leads us to believe that the random switching decision should be based on time rather than 'bytes' because we are interested in the download time, not the average capacity itself. Indeed, any algorithm based on bytes or a fixed amount of data will suffer the *curse of bad source peer* in that it has to wait until that amount of data is completely received from the 'bad' source peer. On the other hand, when the decision is based on time, we don't need to wait that long as we can jump out of that source peer after a fixed amount of time (one period).

VI. CONCLUSION

In this paper we have focused on the average download time of each user in a P2P network. With the devastating usage of network resources by P2P applications in the current Internet, it is highly desirable to improve the network efficiency by reducing each user's download time. In contrast to the commonly-held practice focusing on the notion of average capacity, we have shown that both the spatial heterogeneity and the temporal correlations in the service capacity can significantly increase the average download time of the users in the network, even when the average capacity of the network remains the same. We have compared several 'byte-based' (file size based) schemes widely used in practice, including chunk-based file transfer, parallel downloading, as well as their combination, and have shown that all those byte-based schemes are not so effective in reducing the two negative factors that increase the average download time. From our study, it becomes apparent that all P2P algorithms regarding the download time should focus directly on 'time'

rather than on 'bytes', and the notion of average service capacity alone is not sufficient to describe each user's average performance in a P2P network.

REFERENCES

[1] Y. M. Chiu and D. Y. Eun, "Minimizing File Download Time over Stochastic Channels in Peer-to-Peer Networks," in Proceedings of the 40th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, March 2006.

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

[3] X. Yang and G. de Veciana, "Service capacity of peer to peer networks," in Proceedings of IEEE Infocom, Mar. 2004.

[4] 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.

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

[6] Alf Inge Wang and Peter Nicolai Motzfeldt, "Peer2Schedule – an experimental peer-to-peer application to support present collaboration," in International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007), New York, NY, USA, 2007, pp. 418-425.

[7] Loïc Baud and Patrick Bellot, "A purpose adjustable overlay network," in 6th EURO-NF Conference on Next Generation Internet (NGI), , Paris, France, 2010, pp. 1-8.

Kolan Helini M.Tech Computer Science & Engineering from St Mary's Group of Institutions B.Tech from Information Technology Vathsalya Institute of Science & Technology. Currently she is working at Samskruti College of Engineering and Technology has guided many UG & PG students previously she worked in Tirumala Engineering College. Her research areas are Data Mining, Network, and Testing Security.

P. Prashanthi M.Tech Computer Science & Engineering from JNTU Hyderabad B.Tech from Rajiv Gandhi Institute of Technology, Nandyal. Currently she is working at Samskruti College of Engineering and Technology, has guided many UG & PG students. Previously she worked in Narayana College of Engineering and Technology, Nellore. Her research areas include Mobile Computing, Design and Analysis of Algorithms, Complier Design, Computer Architecture, Data Warehousing & Data Mining.

G. Radha Devi M.Tech Computer Science & Engineering from G.Narayanamma Institute of Technology & Science M Sc Physics from St. Pious X Degree and P.G. College. Currently she is working at Samskruti College of Engineering and Technology has guided many UG & PG students. Her research areas are Computer Networks, Network Security, Data Warehousing & Data Mining, Mobile Computing, and Computer Architecture