Riikka Susitaival and Samuli Aalto. Analyzing the file availability and download time in a P2P file sharing system. In Proceedings of the 3rd EuroNGI Conference on Next Generation Internet Networks (NGI 2007), pages 88-95, 2007.

Analyzing the file availability and download time in a P2P file sharing system

Riikka Susitaival
Helsinki University of Technology
Networking Laboratory
PL 3000, 02015 TKK, Finland
Email: riikka.susitaival@tkk.fi

Samuli Aalto
Helsinki University of Technology
Networking Laboratory
PL 3000, 02015 TKK, Finland
Email: samuli.aalto@tkk.fi

***Abstract*— In this paper we study the performance of P2P file sharing systems. We are especially interested in how the division of the file into chunks as well as the used chunk selection policy influence on two performance metrics, named the availability and mean download time of the file. We propose a detailed Markov chain model that allows estimating the mean lifetime of the system and comparing different chunk selection policies exactly. In addition, we use simulations to evaluate more complex systems. Results suggest that splitting the file from one chunk into two chunks improves the performance of the system significantly. Meanwhile, the superiority of a chunk selection policy depends on many factors, such as the arrival rate of the new downloaders and service time distribution.**
Keywords: P2P networking, performance evaluation

I. Introduction

Peer-to-peer (P2P) applications, such as file sharing, have become a significant area of Internet communication in recent years. It has been widely reported that P2P related traffic forms a major part of the total traffic in the Internet and the share is even increasing [1], [2]. Evolution of P2P technologies has been extremely rapid during last 10 years. Whereas Gnutella, Napster and Kazaa were the popular systems in early 2000, they have been succeeded by BitTorrent and its variants in last two years. In addition, popularity of a specific protocol depends heavily on the country. For example, in 2005 BitTorrent was the most used P2P application in Scandinavian, whereas eDonkey was popular in southern Europe [2].

The basic idea of new P2P file sharing protocols, such as BitTorrent [3], is to divide the distributed file into parts, named *chunks*. While downloading the other chunks of the file, the peer uploads the chunks it already has, to other downloaders. By this mechanism the service capacity of the system increases as the number of the downloaders increases, which makes the system scalable.

Measurement studies [4], [5] have shown that the process of sharing a single file is not stationary but can be divided into different phases. In the first *flash crowd* phase the demand for the newly released file, such as a popular movie, is very high. After a couple of days the demand decreases and the flash crowd is followed by a *steady state* in which the demand for the file and the service capacity of the system are in balance. Due to the decentralized manner of the P2P file sharing systems, there are not any guarantees that all the chunks of the file are present in the system over time. If some chunk of the file is missing, the file is not complete anymore. Depending on application, this might or might not be crucial. In this paper we assume the former. This third phase of the system lifetime is called *end phase*.

Some papers have analyzed P2P file sharing systems by stochastic models so far. In early studies Yang and de Veciana divide the analysis of BitTorrent-like system into transient and steady state regimes [6]. The service capacity of the transient regime is studied by a branching process and the steady state by a Markov model. Qiu and Srikant study the performance of the system by a deterministic fluid model in [7]. Their fluid model is supplemented in many other papers, such as [9] and [8]. However, these models mainly assume that all the chunks of the file remain in the system over time. Models that take the integrity aspect into account can be found from [10] and [11]. However, the models are very coarse and thus only indicative. In addition, they are not detailed enough to capture how the peer selection policy influences on the performance of the file sharing system.

As the detailed models to capture the integrity of the file can not be found in earlier papers, in this paper we study first the lifetime of the file sharing process in a P2P application. The disappearance of a single chunk means that the file is not entire anymore and the whole process dies. Especially, we study how the policy in selecting the next chunk for download and division of the file into chunks influence on the mean lifetime of the process.

Second, we study the download time of the total file experienced by the peers. This issue has been widely studied but the earlier analytical models do not capture a) chunk selection and b) file splitting policies. The analyzed chunk selection policies in this paper are:

- Random peer selection. First the peer for downloading is randomly selected among all peers in the system and after that a random chunk from the selected peer's chunk collection is downloaded.
- Rarest chunk first policy. The chunk that has least replicas in the system is first selected for downloading. The peer to upload this chunk is selected randomly among all peers that have that chunk.
- Most common chunk first. The chunk that is most com-

mon is selected for downloading. The peer to upload this chunk is selected randomly among all peers that have the chunk.
- Random chunk policy. First the chunk to be downloaded is selected randomly among all chunks and then a random peer having this chunk is selected.

In this paper we propose a detailed Markov model for evaluating the aforementioned performance metrics in a small system having 1 or 2 chunks. The mean lifetime of the file, that is, the time from the release of the file until the disappearance of it, is estimated by solving the mean absorption time of the Markov model. The mean download time is estimated by solving the state probabilities in a modified Markov model, which is made stable by keeping the original seed in the system. In addition to the Markovian analysis, simulations are performed for more complex systems. We have studied the population dynamics of the P2P file sharing system already in papers [12] and [13]. In those papers we modelled the evolution of the system with a single chunk. In this paper we consider a more complicated system with multiple chunks.

The paper is organized as follows: In Section II we describe the P2P file sharing system we are studying. Section III proposes a detailed Markov model for P2P file sharing system with 1 or 2 chunks. Then in Section IV the lifetime of the system and in Section V the mean download time is studied by the Markov model and simulations. In Section VI we briefly consider the stability issues of the file sharing system. Finally, in Section VII we conclude our paper.

II. System description

In this section we describe the general idea of the system that we are studying. Due to rapid progression of the P2P file sharing protocols and applications, we do not want to restrict ourselves to technical details of any certain protocol, but describe the general idea of those systems. Our meaning is to keep the system description as simple as possible for better understanding the influence of the chunk selection policies on the performance metrics. For that reason some details of the protocols, such as the choking algorithm in BitTorrent, are excluded.

In the system we have three types of peers, *downloaders*, *leechers*, and *seeds*. Downloaders do not have any chunk yet and try only to download the first one. Leechers have already downloaded a part of the chunks and can upload those chunks to other peers in the system. While uploading the chunks, they try to find the rest of the chunks. Finally, peers that have all the chunks are called *seeds*.

Let L be the size of the file under consideration in bytes. The file is divided into blocks, named chunks, which are assumed to be downloaded one after another separately. If we fix the size of chunk to l bytes, the total number of chunks is naturally L/l, denoted by K. New peers requesting the given file are assumed to arrive at the system with rate λ according to the Poisson process.

When new downloader i has arrived, it seeks peer j and a chunk among all available peers and chunks including leechers and seeds according to the chunk selection policy. After the peer has downloaded the first chunk, it seeks a next peer and a next chunk. It can then also upload the chunk it has, to other peers as a leecher, if required. When the leecher has collected all K chunks, the status of the peer changes from a leecher to a seed. We assume that the seeds stay in the system for a random period. Let γ denote the departure rate of a seed. If the seed of the downloader leaves before the download process for a chunk is ended, the downloader has to find a new peer and start to download the chunk again from the beginning.

We denote by μ_d the download rate and by μ_s the upload rate of a peer. The mean download time of a chunk is fixed to be proportional to the chunk size, that is, $\frac{1}{K\mu_d}$. Also the mean upload time of a chunk is proportional to the chunk size, $\frac{1}{K\mu_s}$. If there are many downloaders per one leecher or seed concurrently, we assume that the upload capacity of the peer is divided among downloaders, but the actual number of downloaders per uploading peer is not restricted. We also assume that the time required for finding the peer for exchanging the chunks is negligible as compared to the download time.

III. Markov model for the system with 1 or 2 chunks

A detailed model of the system with multiple chunks described in the previous section is very complicated and hard to solve. On the other hand, more simple models, such as deterministic fluid models, describe only the average behavior of the sharing of the chunks and are thus unable to capture the dynamic nature of the file sharing process including possible unstability and extinction of the system. For making a compromise, we propose a detailed Markov chain model for the P2P file sharing process when the number of the chunks is small, either $K = 1$ or $K = 2$. However, even when the maximum number of the chunks is only two, many characteristics of the system can still be analyzed.

A. One chunk

First we consider the case where the file is not divided into chunks at all and therefore the number of chunks $K = 1$. When the downloaders have downloaded the file, their status change directly from downloaders to seeds. Thus we do not have leechers in the system at all. Let $x(t)$ be the number of downloaders and $y(t)$ be the number of seeds at time t. In the system, if the total download capacity of the downloaders, $\mu_d x(t)$, is smaller than the total upload capacity of the seeds, $\mu_s y(t)$, the downloaders can not use all service capacity provided by the peers. On the other hand, when $\mu_d x(t) > \mu_s y(t)$ the upload capacity of seeds limits the download process. Thus the total service rate of the system is $\min\{\mu_d x(t), \mu_s y(t)\}$. Assuming exponentially distributed download and upload times, we can construct a continuous time Markov chain process of the system, where the state is the pair (x, y) and the transition rate matrix is Q with the

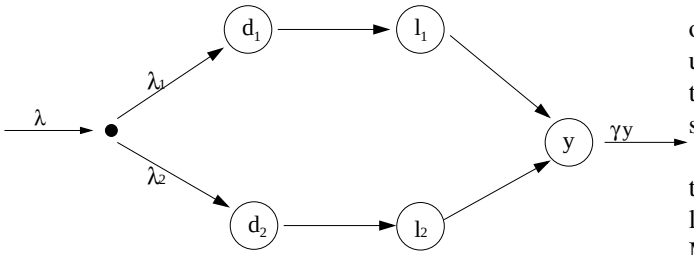

Fig. 1. A model for two chunks.

elements:

$$\begin{aligned} &q((x,y),(x+1,y)) &&= \lambda, \\ &q((x,y),(x-1,y+1)) &&= \min\{\mu_d x, \mu_s y\}, \text{ if } x > 0, \\ &q((x,y),(x,y-1)) &&= \gamma y, \text{ if } y > 0, \end{aligned} \tag{1}$$

where λ is the arrival rate of the new requests for the file in question.

B. Two chunks

Second the file is divided into two chunks, named chunk 1 and chunk 2. We have two types of downloaders; the downloaders that download chunk 1 and the downloaders that download chunk 2. Also the leechers can be divided into two groups; the leechers that have chunk 1 and the leechers that have chunk 2. Let $d_1(t)$ denote the number of the downloaders for chunk 1 and $d_2(t)$ the downloaders for chunk 2 at time t. These peers do not have any chunk. Let $l_1(t)$ and $l_2(t)$ denote the number of the leechers of type 1 or 2, respectively. Finally, let $y(t)$ be the number of the seeds having both chunks 1 and 2 at time t. Let quintet $\mathbf{x} = \{d_1, d_2, l_1, l_2, y\}$ be the state of the system. In Figure 1 we have illustrated the model by a simple flow diagram.

The download time of a chunk depends both on the load of the peer that is uploading to the downloader and on the downloaders' own download capacity. In the analytical model we assume that the maximum possible download rate of a peer for a chunk is $2\mu_d$ and the maximum upload rate is $2\mu_s$. The actual upload rate that the downloader receives, is a fair share of the total upload capacity. This share depends both on the number of leechers having the chunk, on the number of the seeds and on the number of peers that are downloading the chunk concurrently and sharing the capacity. The share of the upload capacity of leechers and seeds per downloader for chunk 1 is denoted by $\phi_1(\mathbf{x})$ and can be written as

$$\phi_1(\mathbf{x}) = \left(\frac{l_1}{d_1 + l_2} + \frac{y}{d_1 + d_2 + l_1 + l_2}\right) 2\mu_s. \tag{2}$$

Similarly, let $\phi_2(\mathbf{x})$ be the shared upload capacity for chunk 2:

$$\phi_2(\mathbf{x}) = \left(\frac{l_2}{d_2 + l_1} + \frac{y}{d_1 + d_2 + l_1 + l_2}\right) 2\mu_s. \tag{3}$$

Note that the model is idealistic in the sense that in a real system the downloader is hardly able to utilize all upload capacity of the leechers and the seeds due to mismatch of the peers. In addition, the leechers and seeds probably upload the parts of some *other* file concurrently. However, these assumptions have to be made to keep analytical model solvable.

Assuming exponentially distributed download and upload times, the evolution of the downloaders, different type of leechers and the seeds can be described by a five-dimensional Markov chain, whose state is $\mathbf{x} = (d_1, d_2, l_1, l_2, y)$ and the transition rate matrix is Q with the following elements:

$$\begin{aligned} &q(\mathbf{x}, \mathbf{x} + e_{d_1}) = \lambda_1(\mathbf{x}), \\ &q(\mathbf{x}, \mathbf{x} + e_{d_2}) = \lambda_2(\mathbf{x}), \\ &q(\mathbf{x}, \mathbf{x} - e_{d_1} + e_{l_1}) = \min\{2\mu_d d_1, \phi_1(\mathbf{x}) d_1\}, \\ &q(\mathbf{x}, \mathbf{x} - e_{d_2} + e_{l_2}) = \min\{2\mu_d d_2, \phi_2(\mathbf{x}) d_2\}, \\ &q(\mathbf{x}, \mathbf{x} - e_{l_1} + e_y) = \min\{2\mu_d l_1, \phi_2(\mathbf{x}) l_1\}, \\ &q(\mathbf{x}, \mathbf{x} - e_{l_2} + e_y) = \min\{2\mu_d l_2, \phi_1(\mathbf{x}) l_2\}, \\ &q(\mathbf{x}, \mathbf{x} - e_y) = \gamma y, \end{aligned} \tag{4}$$

where e_i is a vector, the elements of which are zeros, except element i, which is 1. The arrival rate of type 1 and 2 downloaders, denoted by λ_1 and λ_2, respectively, depends on both the chunk selection policy and the state of the system. The arrival rates for each policy are explained next.

Random peer selection: In the random peer selection policy (denoted also by RND peer policy), when a downloader starts to download the first chunk, it selects a peer randomly among all peers in the system. The selected peer is a leecher with chunk 1 with probability $\frac{l_1}{l_1+l_2+y}$, a leecher with chunk 2 with probability $\frac{l_2}{l_1+l_2+y}$ and a seed with probability $\frac{y}{l_1+l_2+y}$. The seeds have both chunks 1 and 2, so if the downloader selects a seed for chunk exchange, the probability to get chunk 1 or chunk 2 is equal. Therefore, the total rate to download first chunk 1 is

$$\lambda_1(\mathbf{x}) = \frac{l_1 + \frac{y}{2}}{l_1 + l_2 + y}\lambda.$$

Correspondingly,

$$\lambda_2(\mathbf{x}) = \frac{l_2 + \frac{y}{2}}{l_1 + l_2 + y}\lambda.$$

Rarest chunk first selection: In some P2P protocols, such as BitTorrent, peers do not select the chunks for download randomly, but based on the rareness of chunks. The rarest chunk is downloaded first to ensure the uniform distribution of the chunks in the system and to avoid the last chunk problem. This policy (denoted later by RF) requires that peers keep up updated information about the rareness of different chunks.

In the model, if $l_1 < l_2$, chunk 1 is rarest and new peers start to download chunk 1. In that case $\lambda_1(\mathbf{x}) = \lambda$ and $\lambda_2(\mathbf{x}) = 0$. On the other hand, if $l_1 > l_2$, all downloaders start to download chunk 2 and $\lambda_1(\mathbf{x}) = 0$ and $\lambda_2(\mathbf{x}) = \lambda$. If $l_1 = l_2$, obviously $\lambda_1(\mathbf{x}) = \lambda_2(\mathbf{x}) = 0.5\lambda$.

Most common chunk first: The most common chunk first policy (denoted by MCF) is just the opposite of the rarest chunk selection. The arrival rates for that policy can easily be obtained from the rarest chunk first case by changing the directions of the inequalities. Thus, if $l_1 > l_2$, then $\lambda_1(\mathbf{x}) = \lambda$ and $\lambda_2(\mathbf{x}) = 0$; if $l_1 < l_2$, then $\lambda_1(\mathbf{x}) = 0$ and $\lambda_2(\mathbf{x}) = \lambda$; and if $l_1 = l_2$, then $\lambda_1(\mathbf{x}) = \lambda_2(\mathbf{x}) = 0.5\lambda$.

Random chunk selection: In the random chunk selection (denoted by RND chunk policy), first a peer itself selects the next chunk randomly among the chunks it does not have yet, and after that it selects a peer that has the selected chunk. In the case of two chunks the arrival rates for different type of downloaders are thus $\lambda_1(\mathbf{x}) = \lambda_2(\mathbf{x}) = 0.5\lambda$.

IV. Lifetime of the file sharing process

In this section we study the lifetime of the file sharing process, that is, how long the file is available from the release until disappearance of it. We do not assume that there is always a seed keeping the file entire in the system. If one of the chunks is missing, i.e. there are no seeds in the system and the chunk is not included in any chunk collection held by other leechers, the file is not complete anymore. In this case the sharing process of the given file dies. Our detailed Markov chain model described in the previous section gives tools to evaluate the lifetime of the process by calculating the absorbtion time of the system.

When we consider the one chunk model, the seeds are the only peers that provide the file and thus all the states (x, y) with $y = 0$ in the Markov chain are absorbing states. On the other hand, in the system with two chunks, if there are no seeds in the system and leechers together do not have the complete file, the system dies. Thus the states $\mathbf{x} = (d_1, d_1, l_1, l_2, y)$ with $y = 0 \cap (l_1 = 0 \cup l_2 = 0)$ are absorbing. Given the transition matrix Q, the mean time to absorption can be solved from the linear system of equations determined by a familiar Markovian recursion (see formula (2) in [12]).

On the left hand side of Figure 2 we show the mean absorption times, i.e. the lifetime of the system, as a function of the average seed departure time $1/\gamma$ for the one chunk model and the two chunk model with various selection policies, when initially there is one seed ($y(0) = 1$) and no downloaders or leechers in the system. We can see that both the division of the file into two pieces and the use of the RF policy improves the file availability since the mean absorption time is longer. However, the differences between the policies are not significant. On the right hand side of Figure 2 the mean absorption time of the system is depicted as a function of arrival rate λ. The result is that the system with a greater demand has also a longer lifetime since there are more leechers keeping the file available. Also the advantages using the RF policy are greater as λ increases.

Next we simulate a similar system for obtaining results for greater number of chunks as well. The event-based simulation corresponds the detailed description of the file sharing system in Section II. The difference to analytical model is that the pairs of peers exchanging the chunks are formed. The share of the upload capacity is not calculated from a common "pool", but from the shared capacity of the selected uploader. In addition, in the simulation we do not assume exponential distribution for download times as in Markov model but the download time with full download rate is assumed to be deterministic which better reflects the reality.

The left side of Figure 3 shows the mean lifetime of the file sharing when the number of chunks, K, varies from 1 to 100. The used chunk selection policy is the RND peer. The result is that by dividing the file into two pieces improves the lifetime as compared to sharing the file as one chunk. When use of 10 chunks is compared to two chunks, the difference is smaller. Finally, for small $1/\gamma$, using 10 chunks provides as good result as using 100 chunks. From the results we can conclude that the analytical results for 2 chunks in terms of the mean lifetime are reasonable as compared with real systems having more chunks. In the right side of Figure 3 the chunk selection policies are compared. The simulation result verifies to the result of the analytical model: using the RF chunk selection the mean lifetime is longest.

V. Mean download time

In addition to the mean lifetime of the file sharing process, another performance metric that we are considering is the total download time of the file. The question is how much the download time depends on the number of chunks and the used chunk selection policy. In the previous section we did not assume any seed that is responsible for keeping the file available. For that reason the system finally died out. However, when the mean download time is studied, the analysis of a non-absorbing system, in which the original seed is assumed to keep the file entire, gives more consistent results.

First we study the mean download time of the file sharing system by the analytical Markov model. The model is modified slightly to be recurrent; in the formulas for shared capacities ϕ_1 and ϕ_2 and arrival rates λ_1 and λ_2 the number of seeds y is replaced by $y + 1$. The mean download time T can be directly calculated by solving the the steady state probabilities of the Markov chain and using then Little's formula.

The left side of Figure 4 presents the analytical results for the mean download time as a function of $1/\gamma$. The result is that the division of the file into two chunks decreases the download time substantially. The reason for the decrease is intuitive; when there are two chunks, while downloading the second chunk the peer can upload the first chunk and thus increase the total service capacity of the system. As peer selection policies are compared, the RF policy gives the best performance, and the random peer poorest. However, the difference is very small. In the right side of Figure 4 the download time is studied when arrival rate λ changes. Increase in arrival rate first increases the download time but the system stabilizes after $\lambda > 1$.

In the next figures we study the mean download time for greater values of K by simulations. The left hand side of Figure 5 depicts the mean download time of the whole file as a function of $1/\gamma$ when K varies from 1 to 100 and the chunk selection policy is RND peer. The reference value is $1/\mu_d = 1$, which is the download time with full download rate. We can

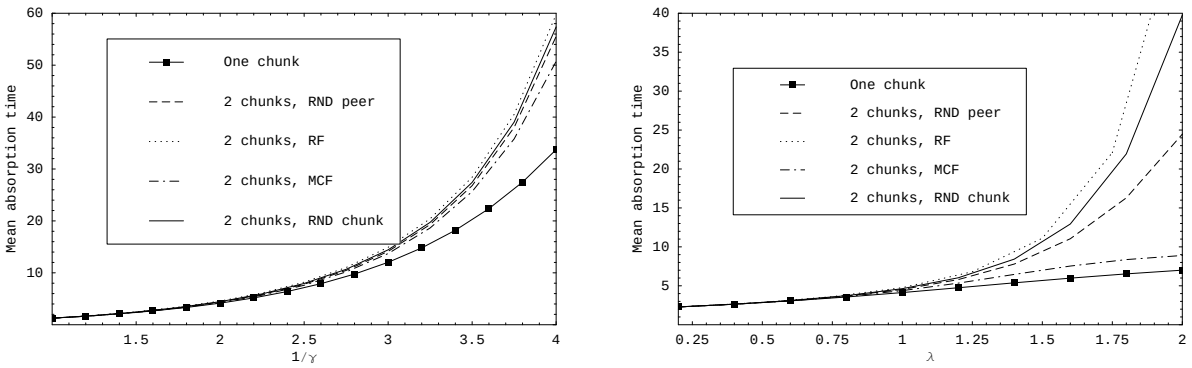

Fig. 2. The mean lifetime as a function of $1/\gamma$ (left hand side, $\lambda = 1$) and as a function λ (right hand side, $\gamma = 1$). Download and upload rates $\mu_d = \mu_s = 1$.

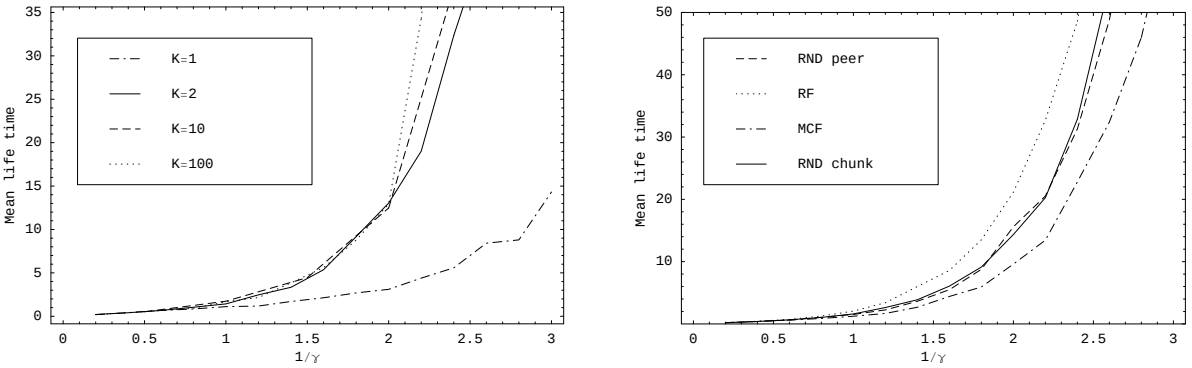

Fig. 3. The mean lifetime as a function of $1/\gamma$. Left side: Random chunk selection, K varies. Right side: $K = 10$, chunk selection policy varies. Download and upload rates $\mu_d = \mu_s = 1$ and $\lambda = 2$.

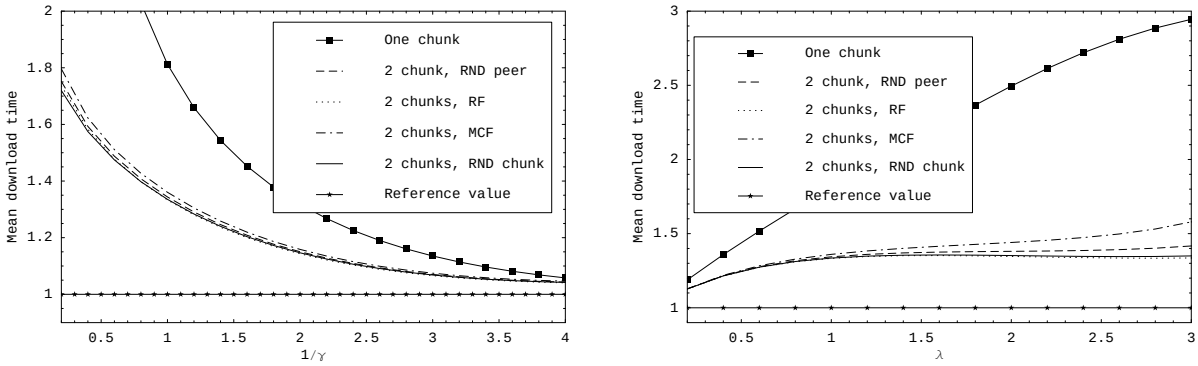

Fig. 4. The mean download time as a function of $1/\gamma$ (left hand side, $\lambda = 1$) and as a function λ (right hand side, $\gamma = 1$). Download and upload rates $\mu_d = \mu_s = 1$.

see that as $1/\gamma$ increases, the mean download time decreases due to increased service capacity. Again, the biggest savings in the download time are done when $K = 1$ changes to $K = 2$. On the right hand side of the same figure the download time is depicted for different chunk selection policies. Interestingly for this parameter combination, the performance is poorest with the RF policy. Some reasons for this phenomenon are given in the end of this section.

In Figure 6 the mean download time is depicted as a function of the arrival rate λ for $K = 2$ (left) and $K = 10$ (right). The results show that the increase in the arrival rate increases the mean download time only in the beginning. The exceptions are the policies RND peer and MFC, which are unstable for $K = 2$ and $\lambda > 3$. Increasing the number of the chunks from 2 to 10 improves the scalability of the system a lot.

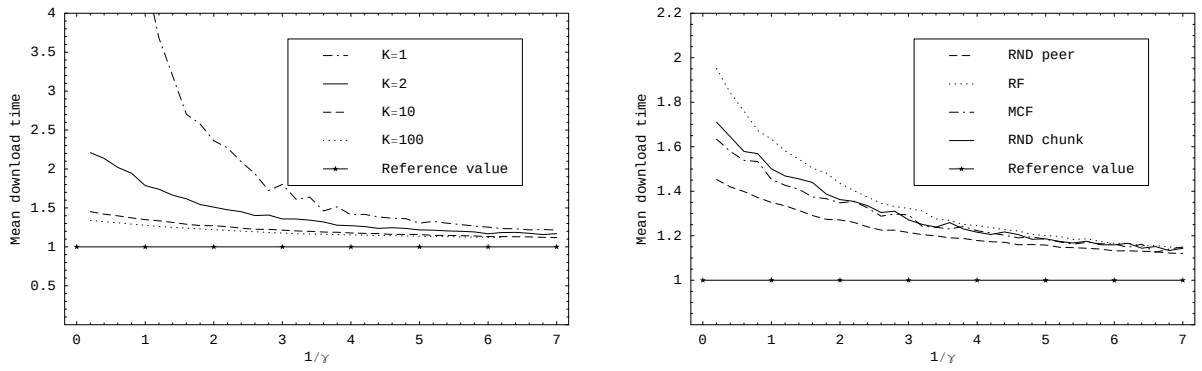

Fig. 5. The mean download time as a function of $1/\gamma$. Left hand side: RND peer policy, K varies. Right hand side: $K = 10$, chunk selection policy varies. Download and upload rates $\mu_d = \mu_s = 1$.

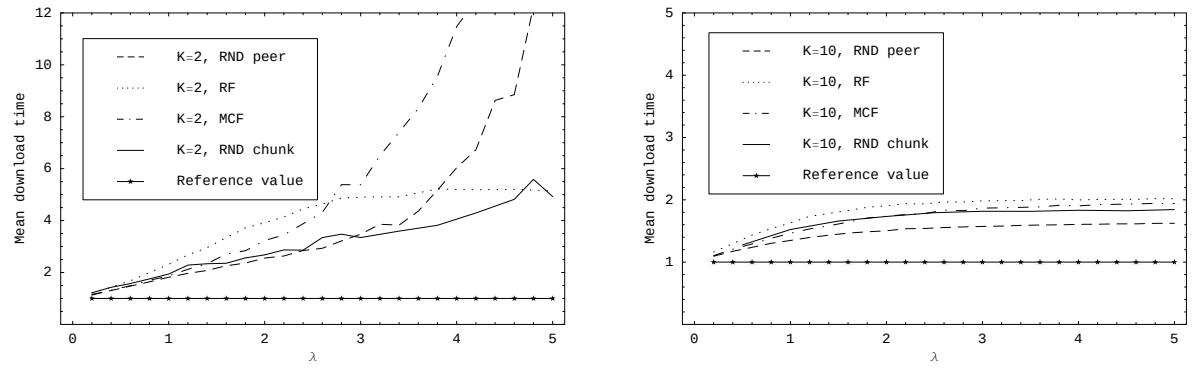

Fig. 6. The mean download time as a function of λ, when $\gamma = 1/2$ and $\mu_d = \mu_s = 1$. Left hand side: K=2, right hand side: K=10.

As regards to the chunk selection policies, from the simulation results we conclude following: First, when the number of the chunks is small like $K = 2$ and the arrival rate of the new peer is high, using the RND peer policy or the MCF policy can lead to poor performance. The reason is that in some situations when the number of type 1 leechers is bigger than type 2, the probability to select type 1 leecher is bigger and the number of type 1 leechers increases until all leechers are of type 1. Then the bottleneck of the system is in downloading of chunk 2 from the small number of seeds. If the RF policy is used, there is a balance between type 1 and 2 leechers and performance is better.

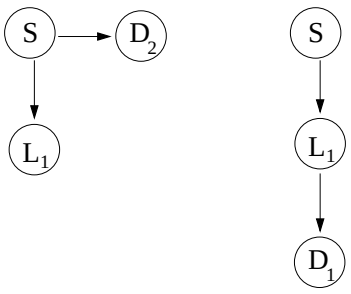

Fig. 7. RF policy (left hand side) and RND peer policy (right hand side).

However, when the arrival rate is low or we increase the number of chunks to 10, for example, the RND peer and MCF policies lead to better performance than RF. The explanation for this can be found from two things: First, the RND peer policy utilizes the seeds and leechers more evenly. Second, in the RF policy the service capacity of the leechers having common chunks probably remain unutilized. A simple scenario is depicted in Figure 7. When a new downloader arrives, according to the RND peer selection, it can download chunk 1 or 2 from the original seed or chunk 1 from the leecher. If the leecher is selected, also its service capacity is utilized. If the rarest chunk is selected (chunk 2 in this case), only the original seed has the chunk and its service capacity is divided and the leechers' service capacity is unused. Thus the resource sharing is not optimal in the latter peer selection policy.

The fourth policy, RND chunk policy, is a compromise of the other policies. For small λ it gives better results than RF and for great λ it is better than RND peer or MCF. The benefit of this approach is that we do not have to find out what chunk is rarest but we can just select a random one.

VI. Stability considerations

In this section we shortly consider the stability of the file sharing systems with different peer selection policies. In

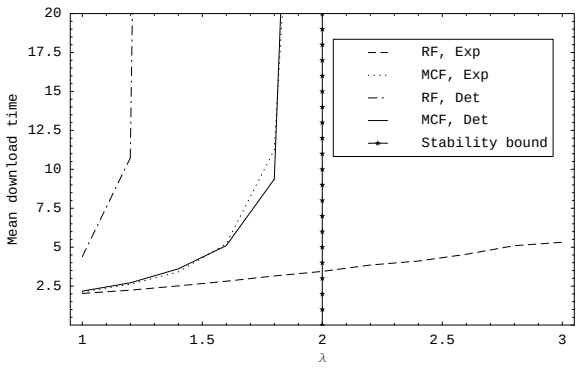

Fig. 8. The mean download time as a function of λ.

earlier studies of BitTorrent-like systems, such as [7] and [8], the stability bounds of the system are derived from the deterministic fluid model. The fluid model corresponding to our Markov model is:

$$\begin{aligned}
\frac{dd_1(t)}{dt} &= \lambda_1(\mathbf{x}(t)) - \min\{2\mu_d d_1(t), \phi_1(\mathbf{x}(t))d_1(t)\},\\
\frac{dd_2(t)}{dt} &= \lambda_2(\mathbf{x}(t)) - \min\{2\mu_d d_2(t), \phi_2(\mathbf{x}(t))d_2(t)\},\\
\frac{dl_1(t)}{dt} &= \min\{2\mu_d d_1(t), \phi_1(\mathbf{x}(t))d_1(t)\}\\
&- \min\{2\mu_d l_1(t), \phi_2((\mathbf{x}(t))l_1(t)\},\\
\frac{dl_2(t)}{dt} &= \min\{2\mu_d d_2(t), \phi_2(\mathbf{x}(t))d_2(t)\}\\
&- \min\{2\mu_d l_2(t), \phi_1(\mathbf{x}(t))l_2(t)\},\\
\frac{dy_1(t)}{dt} &= \min\{2\mu_d l_1(t), \phi_2(\mathbf{x}(t))l_1(t)\}\\
&+ \min\{2\mu_d l_2(t), \phi_1(\mathbf{x}(t))l_2(t)\} - \gamma y(t),
\end{aligned} \tag{5}$$

where $\mathbf{x}(t) = \{d_1(t), d_2(t), l_1(t), l_2(t), y(t)\}$. In the model the arrival rates depend on the state of the system and the chunk selection policy. However, due to symmetry of the system, we can distinguish two different resulting scenarios for the steady state values of $d_1(t)$, $d_2(t)$, $l_1(t)$, $l_2(t)$ and $y(t)$, denoted by $\overline{d}_1$, $\overline{d}_2$, $\overline{l}_1$, $\overline{l}_2$ and $\overline{y}$.

Let us first consider the MCF policy. If the number of the leechers for another chunk, say chunk 1, is greater than the number of the leechers for chunk 2 ($l_1(t) > l_2(t)$), the downloaders download only chunk 1 and the number of leechers for chunk 1 increases until there are only type 1 downloaders and type 1 leechers. Thus in the steady state $\lambda_1 = \lambda$, $\lambda_2 = 0$, $\overline{d}_2 = 0$ and $\overline{l}_2 = 0$. Assuming that $\mu_d = \mu_s = \mu$, the steady state solution for the differential equations (5) exists (and the system is stable) in the following two cases:

- $\gamma < 2\mu$ and $\lambda < \infty$.
- $\gamma \geq 2\mu$ and $\lambda < \dfrac{1}{\dfrac{1}{2\mu} - \dfrac{1}{\gamma}}$.

In the second scenario we consider the RF policy. If $l_1(t) > l_2(t)$ for some t, downloaders download only chunk 2 and the number of leechers for chunk 2 increases until it is same as the l_1. When $l_1 = l_2$, half of the downloaders download chunk 1 and half chunk 2. Thus in the steady state $\lambda_1 = \lambda_2 = 0.5\lambda$ and $\overline{d}_1 = \overline{d}_2$ and $\overline{l}_1 = \overline{l}_2$. Contrary to the previous scenario, the steady state exists and the system is stable for all λ, μ and γ. This result is the same as the corresponding results from papers [7] and [8]; the average download time under RF policy approaches a finite limit as the arrival rate λ increases.

Next we compare the above stability results of the MCF and RF policies to the simulations of the corresponding Markov chain model. We use simulations instead of analytical results to avoid the effect of the truncated state space on the download time. Let $\mu = 1$ and $\gamma \to \infty$. According to the stability analysis, the MCF policy should be stable, if $\lambda < 2$, and RF for all λ. In Figure 8 we have plotted the mean download time as a function of λ. In the first two lines the download time is exponentially distributed as is the assumption in the Markov model. The MCF policy seem to be unstable when λ is close to 2 and download time of RF policy increases only linearly indicating correctness of the stability analysis. However, interesting is to note that when the service time distribution is changed to deterministic, the MCF policy behaves similarly as in the previous case, but the RF policy is unstable for very small values of λ. The conclusion from this simple simulation scenario is that the service time of the P2P file sharing system is not only sensitive to the chunk selection policy but also to the download time distribution. For that reason the deterministic fluid model is not sufficient to evaluate the stability.

VII. Conclusions

In this paper we have studied the file availability and the download time in a general, BitTorrent-like P2P file sharing system. We have proposed a detailed Markov model which allows us to estimate the lifetime of the file sharing process. In addition to this, we used also simulations both to study more complex systems and verify the analytical model. Even though our model and simulations are limited in some sense, we can conclude that dividing the file into chunks improves the performance of the P2P file sharing system in terms of file availability and total download time. The biggest improvements are done when the file is divided from one piece into two pieces. Whether the number of the chunks is 10 or 100 does not have so significant influence on the performance anymore. One can assume that increasing the number of the chunks increases also the overhead costs and delay. Taking this into account, our study suggests that the optimal number of chunks appears to be relatively small.

In addition to the division of the file into chunks, we have studied the influence of the peer/chunk selection policy on the performance of the system. Our conclusion is that the

superiority of a policy depends on the network parameters. The rarest first (RF) peer selection policy improves the lifetime of the system especially when λ is great. However, as regard to the download time, our conclusion is that for small λ or great K, the RND peer policy provides the smallest download time, whereas for great λ and small K the RF policy is the best.

In the end of the paper we have also shortly studied the stability of the file sharing. Simulations showed that, among others, the stability of the system may be sensitive to the service time distribution, and simple deterministic models presented in the literature are not necessarily adequate to find stability bounds. Our aim in the future is to find out better approaches to study stability issues.

REFERENCES

[1] T. Karagiannis, A. Broido, N. Brownlee, kc claffy, M. Faloutsos, Is P2P dying or just hiding?, in Globecom, 2004.

[2] http://www.cachelogic.com/home/pages/research/p2p2005.php

[3] B. Cohen, Incentives Build Robustness in BitTorrent, 2003, in Proc. of First Workshop on Economics of Peer-to-Peer Systems, June 2003, http://www.bittorrent.com/bittorrentecon.pdf.

[4] M. Izal, G. Uvroy-Keller, E.W. Biersack, P.A. Felber, A.Al Hamra, and L. Garcés-Erice, Dissecting BitTorrent: Five Months in a Torrent's Lifetime, in PAM, 2004.

[5] J.A. Pouwelsem, P. Garbacki, D.H.J. Epema, H.J. Sips, The BitTorrent P2P File-sharing system: Measurements and analysis, in IPTPS, 2005.

[6] X. Yang and G. de Veciana, Service Capacity of Peer to Peer Networks, in INFOCOM 2004.

[7] D. Qiu and R. Srikant, Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks, in SIGCOMM 2004.

[8] B. Fan, D-M Chiu, and J. Lui, Stochastic Differential Equation Approach to Model BitTorrent-like P2P Systems, In IEEE ICC, 2006.

[9] L. Massoulié and M. Vojnović, Coupon replication Systems, in SIGMETRICS, 2005.

[10] Y. Tian, D. Wu, K. and W. Ng, Modeling, Analysis and Improvement for BitTorrent-Like File Sharing Networks, in INFOCOM 2006.

[11] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, Measurement, Analysis and Modeling of bitTorrent-like Systems, in Internet Measurement Conference of USENIX Association, 2005.

[12] R. Susitaival, S. Aalto, and J. Virtamo, Analyzing the dynamics and resource usage of P2P file sharing by a spatio-temporal model, in P2P-HPCS'06 in conjuction with ICCS'06, pp. 420–427, 2006.

[13] R. Susitaival and S. Aalto, Modelling the Population Dynamics and the File Availability in a BitTorrent-like P2P System with Decreasing Peer Arrival Rate, in IWSOS, 2006.