Segmented P2P Video-on-Demand: Modeling and Performance

Samuli Aalto, Pasi Lassila
TKK Helsinki University of Technology
Email: firstname.lastname@tkk.fi

Petri Savolainen, Sasu Tarkoma
Helsinki Institute for Information Technology
Email: firstname.lastname@hiit.fi

Abstract—**In a segmented peer-to-peer video-on-demand system the video file is split into a number of segments and the downloading of the video file proceeds in a more or less sequential manner from one segment to another (i.e., in stages). We present an analytical fluid model for such systems. Notably, for this model we derive an explicit condition when the system has a unique positive steady-state solution and that the viewing quality is acceptable. The analytical results are complemented with extensive simulations from the corresponding stochastic model, as well as traces from a more realistic BitTorrent simulator.**

I. Introduction

Peer-to-Peer (P2P) protocols such as BitTorrent are widely used in file sharing, and they are also gaining popularity in video-on-demand (VoD) and live streaming applications. The collaboration between the peers in P2P systems enables scalability to large numbers of simultaneous users.

We consider the specific case of VoD, also referred to as on-demand streaming of stored media [1]. A well-known example of a VoD application is YouTube where users requesting a video are served on a client-server basis. A key question is then how much and under what conditions can a VoD service benefit from P2P assisted operation?

In P2P VoD, the whole stored media file needs to be retrieved (cf., file sharing) at such a rate which allows, as soon as possible, the pieces to be played back in sequential order at the media playback rate (cf., live streaming). Thus, there are two phases, the *transfer* and the *playback* phases, which are partly overlapping. If the retrieval rate is sufficient, the playback phase extends beyond the transfer phase.

To achieve the above objectives, the downloading mechanism of the P2P VoD system must be based on a roughly sequential download of the pieces of the file. However, it does not need to be strictly sequential as the VoD application allows more degrees of freedom in the downloading than live streaming, as shown in [1]. Experimental P2P systems have been developed for VoD based on windowed BitTorrent [2], where an adaptive window size is used to speed up the downloading. Also, combining P2P and network coding for VoD has been proposed, e.g., in [3], [4], where the file is split into segments and network coding is used over the pieces in a segment. To make the dissemination of the video file efficient, peers share data within multiple segments.

We develop a fluid model for the system where the original file is split into multiple segments, each consisting of several pieces, and we assume that the downloading proceeds sequentially in stages. This is a conservative assumption as realistic systems [3], [4] can download multiple segments simultaneously. The model consists of differential equations describing the time-dependent evolution of the mean number of leechers in each stage and the mean number of seeds. An explicit condition is derived for guaranteeing that the downloading rate exceeds the viewing rate, i.e., the performance is acceptable. The condition characterizes, e.g., how the number of segments or the number of permanent seeds improves the system, and allows us to assess the gain obtained by using a segmented P2P approach over a traditional client-server approach. We validate the results by comparing them against results from a stochastic model, where the approximations used in the fluid model are not present. Additionally, we compare the accuracy against traces from a segmented BitTorrent simulator.

The paper is organized as follows. Section II discusses the related work. The modeling framework and the fluid model are presented in Section III, followed by the steady state-analysis in Section IV. The stochastic model used for validating the fluid model is defined in Section V. The synthesis combining our insights on the fluid model and the stochastic model are in Section VI. Section VII includes the numerical validations, and conclusions are given in Section VIII.

II. Related work

P2P *live* streaming has been analyzed recently in many papers, including the impact of network coding, see [5], [6], [7]. However, our objective is to consider the case of VoD, or on-demand streaming.

Qiu and Srikant [8] develop a simple fluid model to analyze the performance of a BitTorrent-like P2P *file sharing* system under a steady flow arrival scenario. Their model describes the time-evolution of the system by differential equations. The global stability of this fluid model is proved in [9]. Extension of [8] to heterogeneous users has been given in [10], [11].

Inspired by [8], Parvez et al. [1] develop fluid models to analyze the performance of a BitTorrent-like P2P *on-demand streaming*. They only consider an upload constrained system. Like Qiu and Srikant, Parvez et al. use the parameter η to capture the joint effects of the piece selection policy, the number of downloading connections, and the number of pieces. Both conclude that $\eta \approx 1$ at least for “most scenarios of interest”. However, while the assumption that $\eta \approx 1$ is plausible for file sharing systems, it may not be the case for on-demand-streaming. For example, simulations with realistic P2P VoD systems in [2] have shown that the efficiency of the piece exchange is limited by the used windowing mechanism. The Parvez et al. model also totally ignores the playback phase (which is an essential feature of any VoD system) and simply assumes that any seed, whether it has played back the media file or not, departs with rate γ.

Like [1], our fluid model for an on-demand streaming system is inspired by the file sharing application model of [8]. However, unlike [1],

- Our model does not include the *implicit* efficiency parameter η but we model the efficiency of the piece exchange *explicitly* by taking into account the different stages of the leechers.
- Our model does not include the *arbitrary* departure parameter γ but we include the playback phase explicitly in our departure model. We believe that this more accurately reflects what can be expected from users in a real VoD application.
- Our model is not restricted to the upload-constrained case, but we allow any positive values of the download and upload rates, respectively.

In addition, we generalize the previous models by allowing for a finite number of permanent seeds.

III. Deterministic fluid model

In this section, we present the general modeling framework and the deterministic fluid model of the P2P VoD system. A stochastic counterpart of this model is given later in Section V.

A. Modeling framework

Let m denote the size of the video file to be shared (in bits). New peers arrive at rate λ (arrivals per time unit on average). Each peer is connected to the network over an asymmetric access link (e.g., an ADSL link) with a download capacity of d (bits per time unit) and upload capacity u (bits per time unit). The corresponding download and upload rates are: $c = d/m$ and $\mu = u/m$ (file transfers per time unit).

The life span of a peer consists of two phases, the file *transfer phase* and the *playback phase*, which are partly overlapping. The first part of the transfer phase (called the startup delay) lasts until there are sufficiently pieces in the playback buffer to start the playback phase. After the startup delay, the video is played back at a constant rate w (bits per time unit) so that the total playback time is equal to $z = m/w$. Video transfer and playback proceed in parallel until the entire video is transferred. It is desirable that the transfer rate is greater than the playback rate so that the video can be played back without any breaks or delays. In such a case, the playback phase still continues after the transfer phase. On the other hand, if the transfer rate is less than the playback rate, the playback phase ceases at the same time as the transfer phase.

In a segmented system, the file is divided into I segments with each segment i consisting of a fixed number of pieces (or chunks). The amount of data in a typical segment corresponds to a few tens of seconds of data in the original video stream. Peers download pieces from other peers segment-by-segment. This is a conservative assumption as real systems can optimize the downloading over segments, see, e.g., [3], [4]. Thus, the transfer phase consists of I consecutive stages. A peer is said to be in stage i when it is currently downloading segment i. Since the size of any single segment is m/I, a download-constrained peer completes each stage i with rate Ic.

We assume that the startup delay is negligible, which is justified by the fact that the video file sizes are typically very large compared to the size of the playback buffer. In addition, we assume that

$$d > w, \tag{1}$$

which is a necessary requirement for a VoD system in order that the video can be played back with good quality, since the transfer rate for a downloading peer is always upper bounded by its own access link rate. It follows from (1) that

$$z > 1/c. \tag{2}$$

During the transfer phase, the peer is called a *leecher*. According to the fundamental P2P principle, leechers help each other. However, due to the character of the VoD application, only the leechers that are in a higher stage $j \geq i$ can help a stage-i leecher. In fact, we make a slightly conservative assumption that only those leechers that are in a *strictly* higher stage $j > i$ can help a stage-i leecher.

An altruistic leecher becomes a *seed* as soon as its own transfer phase is completed. Let $\zeta \in [0,1]$ denote the fraction of altruistic peers. Non-altruistic peers are assumed to leave the system immediately after they have completed their own file transfer, while altruistic peers stay in the system until the end of the playback phase but depart immediately thereafter, which is a worst case scenario. In addition to the non-permanent seeds, we allow a number of original seeds, say k, that stay in the system permanently.

B. Fluid model

Let $x_i(t)$ denote the number of stage-i leechers at time t. In addition, let $x(t) = x_1(t)+\ldots+x_I(t)$ and $y(t)$ denote the total number of leechers and non-permanent seeds, respectively, at time t.

The number of stage-1 leechers, $x_1(t)$, increases with rate λ. On the other hand, it decreases with the rate at which the first segment becomes complete for any stage-1 leecher. Assuming that there are no bottlenecks in the core network, this completion rate $\phi_1(t)$ is determined by the minimum of the total download rate and the fair share of the upload rate that the stage-1 leechers get,

$$\phi_1(t) = \min\{\phi_1^d(t), \phi_1^u(t)\}, \tag{3}$$

where

$$\phi_1^d(t) = Icx_1(t)$$

and

$$\phi_1^u(t) = I\mu x_1(t)\left(\frac{x_2(t)}{x_1(t)} + \frac{x_3(t)}{x_1(t)+x_2(t)} + \ldots + \frac{x_I(t)}{x_1(t)+\ldots+x_{I-1}(t)} + \frac{y(t)+k}{x_1(t)+\ldots+x_I(t)}\right).$$

Thus, we have

$$x_1'(t) = \lambda - \phi_1(t). \tag{4}$$

Since the segments are completed in a sequential order, the number of stage-i leechers, $x_i(t)$, increases with the completion rate $\phi_{i-1}(t)$ of stage $i-1$. On the other hand, it decreases with rate $\phi_i(t)$,

$$\phi_i(t) = \min\{\phi_i^d(t), \phi_i^u(t)\}, \tag{5}$$

where

$$\phi_i^d(t) = Icx_i(t)$$

and

$$\phi_i^u(t) = I\mu x_i(t)\left(\frac{x_{i+1}(t)}{x_1(t)+\ldots+x_i(t)} + \ldots + \frac{x_I(t)}{x_1(t)+\ldots+x_{I-1}(t)} + \frac{y(t)+k}{x_1(t)+\ldots+x_I(t)}\right).$$

Thus, we have

$$x_i'(t) = \phi_{i-1}(t) - \phi_i(t). \tag{6}$$

As mentioned above, an altruistic leecher becomes a seed after the final stage I, which completes the whole file transfer phase. Thus, the number of non-permanent seeds, $y(t)$, increases with rate $\zeta\phi_I(t)$ with

$$\phi_I(t) = \min\{\phi_I^d(t), \phi_I^u(t)\}, \tag{7}$$

where

$$\phi_I^d(t) = Icx_I(t)$$

and

$$\phi_I^u(t) = I\mu x_I(t)\frac{y(t)+k}{x_1(t)+\ldots+x_I(t)}.$$

In (3), (5) and (7) it is indeed assumed that upload capacity is shared fairly between the stages, which may not be optimal.

Characterizing the rate at which seeds depart from the system is more difficult and requires some approximations given below. We base our model on *initially* making the optimistic assumption that the transfer rate is sufficient so that the video can be played back without any breaks or delays. Since we also assume that the startup delay is negligible and altruistic peers leave the system as soon as they complete the playback phase, this fixes the total time that such a peer spends in the system to be equal to z. Now we make an *approximative* assumption that the system is quasi-stationary at every point in time t. Then by Little's result, the average time that an altruistic peer stays as a seed equals $z - x(t)/\lambda$. Thus, the total departure rate of seeds becomes $y(t)/(z-x(t)/\lambda)$ which implies that

$$y'(t) = \zeta\phi_I(t) - y(t)/(z - x(t)/\lambda). \tag{8}$$

There are no a priori guarantees that the difference $z - x(t)/\lambda$ stays positive. If this is not the case, the differential equation system behaves in an unstable manner, which can be interpreted as a sign of problems in the playback quality of the P2P VoD application.

All in all, our fluid model is as follows:

$$\begin{cases} x_1'(t) = \lambda - \phi_1(t), \\ x_i'(t) = \phi_{i-1}(t) - \phi_i(t),\ i = 2,\ldots,I, \\ y'(t) = \zeta\phi_I(t) - y(t)/(z - x(t)/\lambda). \end{cases} \tag{9}$$

IV. Steady-state analysis

In principle, model (9) allows us to explicitly solve the equilibrium of the system by setting $x_i'(t) = y'(t) = 0$ (for all i) and solving for the corresponding equilibrium values $\bar{x}_1,\ldots,\bar{x}_I$ and $\bar{y}$,

$$\begin{cases} \lambda - \phi_1 = 0, \\ \phi_{i-1} - \phi_i = 0, \quad i = 2,\ldots,I, \\ \zeta\phi_I - \bar{y}/(z - \bar{x}/\lambda) = 0, \end{cases} \tag{10}$$

where ϕ_i refers to the equilibrium value of $\phi_i(t)$. From this, we get the following immediate results for the equilibrium values:

$$\begin{cases} \phi_i = \lambda \quad \text{for all } i = 1,\ldots,I, \\ \bar{y} = \zeta\lambda(z - \bar{x}/\lambda). \end{cases} \tag{11}$$

Equilibrium values $\bar{x}_1,\ldots,\bar{x}_I$ and $\bar{y}$ are *admissible* if they are strictly positive. Note that (a priori) there is no guarantee about the existence or uniqueness of an admissible equilibrium solution. As explained below, the complete analysis is very challenging, but we have found some important results that are valid for any I.

We first show several structural properties of the steady-state solution which simplifies the general solution significantly. After that, the steady-state solution is characterized more explicitly.

A. Structural properties

Due to the min-operation in (3), (5), and (7), we need to separately consider whether the stages are download or upload constrained at the equilibrium. Stage i is said to be *download constrained* if there are admissible equilibrium values satisfying

$$\phi_i = \phi_i^d = Ic\bar{x}_i. \tag{12}$$

Correspondingly, stage i is said to be *upload constrained* if there are admissible equilibrium values satisfying

$$\phi_i = \phi_i^u = I\mu\bar{x}_i\left(\frac{\bar{x}_{i+1}}{\bar{x}_1+\ldots+\bar{x}_i} + \ldots + \frac{\bar{x}_I}{\bar{x}_1+\ldots+\bar{x}_{I-1}} + \frac{\bar{y}+k}{\bar{x}_1+\ldots+\bar{x}_I}\right). \tag{13}$$

Since there are I stages, this leads (a priori) to 2^I different cases, which would make the analysis extremely laborious. However, as we show below, it is possible to rule out a majority of them so that the number of possible cases increases only linearly as a function of I, which significantly simplifies the solution.

Proposition 1: If stage i is upload constrained, then also stage $i+1$ is upload constrained. In this case, $\bar{x}_i < \bar{x}_{i+1}$.

Proof: Let $i \in \{1, 2, \ldots, I-1\}$. Assume that stage i is upload constrained, i.e., there are admissible equilibrium values satisfying $\phi_i^d \geq \phi_i^u$. Now it follows from (11) that

$$Ic\bar{x}_i = \phi_i^d \geq \phi_i^u = \phi_i = \lambda. \tag{14}$$

On the other hand, by (13),

$$I\mu\bar{x}_i(a+b) = \phi_i^u = \phi_i = \lambda, \tag{15}$$

where we have used notation

$$a = \frac{\bar{x}_{i+1}}{\bar{x}_1 + \ldots + \bar{x}_i}$$

and

$$b = \frac{\bar{x}_{i+2}}{\bar{x}_1 + \ldots + \bar{x}_{i+1}} + \ldots + \frac{\bar{x}_I}{\bar{x}_1 + \ldots + \bar{x}_{I-1}} + \frac{\bar{y}+k}{\bar{x}_1 + \ldots + \bar{x}_I}$$

By combining equations (14) and (15), we get

$$I\bar{x}_i(c - \mu(a+b)) \geq \lambda - \lambda = 0,$$

implying (since $a > 0$) that

$$c \geq \mu(a+b) > \mu b. \tag{16}$$

Consider now what happens if stage $i+1$ were download constrained with the same admissible equilibrium values. Now it would follow from (12) and (11) that

$$Ic\bar{x}_{i+1} = \phi_{i+1}^d = \phi_{i+1} = \lambda. \tag{17}$$

On the other hand, by (13),

$$I\mu\bar{x}_{i+1}b = \phi_{i+1}^u \geq \phi_{i+1}^d = \phi_{i+1} = \lambda, \tag{18}$$

where we have used the same notation for b as above. By combining equations (17) and (18), we get

$$I\bar{x}_{i+1}(c - \mu b) \leq \lambda - \lambda = 0,$$

implying that

$$c \leq \mu b, \tag{19}$$

which contradicts equation (16). Thus, we conclude that stage $i+1$ must be upload constrained.

Assume now that both stages i and $i+1$ are upload constrained. From equation (15), we deduce that

$$I\mu\bar{x}_i(a+b) = \lambda = I\mu\bar{x}_{i+1}b,$$

which implies that $\bar{x}_i < \bar{x}_{i+1}$ since $a, b > 0$. ■

As an immediate consequence, we get the following result.

Proposition 2: If stage $i+1$ is download constrained, then also stage i is download constrained. In this case, $\bar{x}_i = \bar{x}_{i+1} = \lambda/(Ic)$.

Proof: Let $i \in \{1, 2, \ldots, I-1\}$. Assume that stage $i+1$ is download constrained, i.e., there are admissible equilibrium values satisfying $\phi_{i+1}^d \leq \phi_{i+1}^u$. Now it follows from (11) that

$$Ic\bar{x}_{i+1} = \phi_{i+1}^d = \phi_{i+1} = \lambda \tag{20}$$

so that $\bar{x}_{i+1} = \lambda/(Ic)$. In addition, by Proposition 1, we conclude that also stage i must be download constrained. Finally, it follows again from (11) that

$$Ic\bar{x}_i = \phi_i^d = \phi_i = \lambda, \tag{21}$$

implying that $\bar{x}_i = \lambda/(Ic)$. ■

Proposition 3: If stage i is download constrained and stage $i+1$ is upload constrained, then $\bar{x}_i \leq \bar{x}_{i+1}$.

Proof: Let $i \in \{1, 2, \ldots, I-1\}$. Assume that stage i is download constrained, i.e., there are admissible equilibrium values satisfying $\phi_i^d \leq \phi_i^u$. It follows from (12) and (11),

$$Ic\bar{x}_i = \phi_i^d = \phi_i = \lambda. \tag{22}$$

Assume then that stage $i+1$ is upload constrained with the same admissible equilibrium values. Now it follows from (12) and (11) that

$$Ic\bar{x}_{i+1} = \phi_{i+1}^d \geq \phi_{i+1}^u = \phi_{i+1} = \lambda. \tag{23}$$

By combining equations (22) and (23), we get

$$Ic\bar{x}_i = \lambda \leq Ic\bar{x}_{i+1},$$

which implies that $\bar{x}_i \leq \bar{x}_{i+1}$. ■

It follows from Propositions 1, 2, and 3 that for any admissible equilibrium value vector $(\bar{x}_1, \ldots, \bar{x}_I, \bar{y})$ there exists a "switch-over index" $i^* \in \{0, 1, \ldots, I\}$ such that the stages $1, \ldots, i^*$ are download constrained and the rest $(i^*+1, \ldots, I)$ are upload constrained. In addition,

$$\bar{x}_1 = \ldots = \bar{x}_{i^*} \leq \bar{x}_{i^*+1} < \ldots < \bar{x}_I.$$

B. Steady-state solution

The case that all stages are download constrained can be characterized explicitly in a closed form, see Proposition 4 below.

Proposition 4: There is a unique admissible equilibrium solution $(\bar{x}_1, \ldots, \bar{x}_I, \bar{y})$ for which all stages are download constrained if and only if

$$\frac{1}{\mu} \leq \zeta\left(z - \frac{1}{c}\right) + \frac{k}{\lambda}. \tag{24}$$

In this case, the equilibrium values are

$$\begin{cases} \bar{x}_i = \dfrac{\lambda}{Ic} \quad \text{for all } i = 1, \ldots, I, \\ \bar{x} = \dfrac{\lambda}{c}, \\ \bar{y} = \zeta\lambda\left(z - \dfrac{1}{c}\right). \end{cases} \tag{25}$$

Proof: 1° Assume that all stages are download constrained. Let $i \in \{1, 2, \ldots, I\}$. Since stage i is download constrained, it follows from (12) and (11) that

$$\bar{x}_i = \frac{\phi_i}{Ic} = \frac{\lambda}{Ic}.$$

Since this is true for any i, we get

$$\bar{x} = \bar{x}_1 + \ldots + \bar{x}_I = \frac{\lambda}{c},$$

implying, by (11), that

$$\bar{y} = \zeta\lambda\left(z - \frac{\bar{x}}{\lambda}\right) = \zeta\lambda\left(z - \frac{1}{c}\right).$$

Thus, we have proved equation (25). Note that these equilibrium values are admissible (the last one due to (2)).

Since stage I is download constrained, we have

$$\begin{aligned} \phi_I^d \le \phi_I^u \quad &\Longleftrightarrow \quad Ic\bar{x}_I \le I\mu\bar{x}_I\frac{\bar{y}+k}{\bar{x}} \\ &\Longleftrightarrow \quad c \le \mu\frac{\bar{y}+k}{\bar{x}}. \end{aligned}$$

By applying (25), the condition above becomes

$$c \le \mu\zeta c\left(z - \frac{1}{c}\right) + \mu k\,\frac{c}{\lambda},$$

which is clearly equivalent with inequality (24).

2° Assume then that condition (24) is satisfied. As seen from above, only for the values of $(\bar{x}_1, \ldots, \bar{x}_I, \bar{y})$ given in (25), condition (24) implies that $\phi_i^d \le \phi_i^u$ for all i. In other words, all stages are download constrained. In addition, equations (11) are satisfied so that the values of $(\bar{x}_1, \ldots, \bar{x}_I, \bar{y})$ given in (25) constitute an equilibrium solution, which is easily seen to be admissible (due to (2)). ■

Note that condition (24) does not depend at all on the number of stages, I. Also the total numbers, $\bar{x}$ and $\bar{y}$, are independent of I.

For the other cases with at least one upload constrained stage, steady-state analysis can be done explicitly only for $I = 2$. For larger systems, we need to resort to numerical methods. However, the structural properties derived earlier facilitate the analysis considerably. Essentially, instead of needing to consider separately all possible 2^I different combinations of upload/dowload constrained stages (exponential complexity), it is sufficient to consider I different combinations (linear complexity) where one simply needs to examine separately the cases with $i^* = 1, \ldots, I-1$. For each value of i^* one needs to solve an algebraic equation and apply the admissibility constraints. The algorithm to numerically determine the admissible equilibrium values is given in Algorithm 1 below.

Algorithm 1:

1) For each $i^* < I$, find all solutions of the following system of algebraic equations:

$$\begin{cases} 2c\bar{x}_1 = \lambda, \\ \ldots, \\ 2c\bar{x}_{i^*} = \lambda, \\ 2\mu\bar{x}_{i^*+1}\left(\dfrac{\bar{x}_{i^*+2}}{\bar{x}_1 + \ldots + \bar{x}_{i^*+1}} + \ldots + \right. \\ \qquad \left. \dfrac{\bar{x}_I}{\bar{x}_1 + \ldots + \bar{x}_{I-1}} + \dfrac{\bar{y}+k}{\bar{x}_1 + \ldots + \bar{x}_I}\right) = \lambda, \\ \ldots, \\ 2\mu\bar{x}_I\left(\dfrac{\bar{y}+k}{\bar{x}_1 + \ldots + \bar{x}_I}\right) = \lambda, \\ \bar{y} = \zeta\lambda\left(z - \dfrac{\bar{x}_1 + \ldots + \bar{x}_I}{c}\right). \end{cases} \tag{26}$$

2) If a solution exists, verify that the solution is admissible.

We implemented the above routine in Mathematica, which contains a rather efficient solver for algebraic equations. However, even then for a given fixed I the number of possible solutions increases rapidly as i^* becomes small, limiting the efficiency of our numerical approach. However, our numerous numerical experiments (which are omitted here due to lack of space) indicate that a unique admissible solution of (26) is found if and only if

$$\frac{1}{\mu} \ge \zeta\left(z - \frac{1}{c}\right) + \frac{k}{\lambda} \tag{27}$$

and

$$\frac{1}{I\mu} < \frac{k}{\lambda}\left(1 - \frac{I-1}{Icz}\right). \tag{28}$$

Note that (28) is obtained from (26) by assuming that $i^* = I - 1$ (i.e., only last stage is upload constrained) and $\bar{y} = 0$ (no seeds are left in the system).

V. Stochastic model

In addition to a fluid model, we develop a more detailed stochastic model to test the accuracy of the approximative assumptions made for the fluid model. The stochastic model *does not* utilize the quasi-stationarity approximation to estimate the departure rate of altruistic seeds. Instead, the sojourn time is derived from the stochastically evolving service rate of the system. However, the sharing of the upload capacity is still idealized, i.e., the upload capacity of a peer is shared uniformly among the leechers that are in a lower stage.

Let the capital letters $X_i(t)$ and $Y(t)$ refer to the random variables corresponding to the number of stage-i leechers and non-permanent seeds, respectively, at time t. We assume that the new peers arrive according to a Poisson process with rate λ. Let $A(t)$, $C_i(t)$, and $\tilde{D}(t)$ denote the total number of peers arrived, stage-i completions, and seed departures, respectively, until time t. In addition, let $\tilde{A}(t)$ and $\tilde{C}_i(t)$ denote the total number of altruistic peers that have arrived and that have completed stage i, respectively, until time t. We have immediately the following relations:

$$\begin{cases} X_1(t) = A(t) - C_1(t), \\ X_i(t) = C_{i-1}(t) - C_i(t), \quad i = 2, \ldots, I, \\ Y(t) = \tilde{C}_I(t) - \tilde{D}(t). \end{cases}$$

Note that $\tilde{A}(t)$ is derived from $A(t)$ by including each arriving peer in $\tilde{A}(t)$ with probability ζ according to an independent Boolean trial. Below we show that in our model also $C_i(t)$, $\tilde{C}_i(t)$, and $\tilde{D}(t)$ are determined from the arrival process $A(t)$ (and its derivative $\tilde{A}(t)$).

While the playback rate w is deterministic and constant, the transfer rate $R_i(t)$ in stage i is random and time-varying depending on the dynamics of the whole video file sharing system. More precisely, $R_i(t)/(m/I)$ is the stochastic counterpart of $\phi_i(t)$, and can be computed from (5) by replacing $x_i(t)$'s and $y(t)$ by their stochastic versions $X_i(t)$ and $Y(t)$.

A peer stays in the first stage until all pieces of segment 1 have been retrieved. Since the transfer rate $R_1(t)$ in our model is the same for all peers in the first stage, a peer that arrived at time T^a completes the first stage at time T_1^c, where

$$T_1^c = \inf\{t \geq T^a \mid \int_{T^a}^{t} R_1(s)\,ds \geq \frac{m}{I}\}. \tag{29}$$

Vice versa, a peer that completes the first stage at time T_1^c arrived at time T^a, where

$$T^a = \sup\{t \leq T_1^c \mid \int_{t}^{T_1^c} R_1(s)\,ds \geq \frac{m}{I}\}. \tag{30}$$

It follows from (30) that

$$C_1(t) = A(\sup\{t' \leq t \mid \int_{t'}^{t} R_1(s)\,ds \geq \frac{m}{I}\}), \tag{31}$$

The definition of $\tilde{C}_1(t)$ is the same as (31) but applied to $\tilde{A}(t)$. By a similar reasoning, we can generalize these for the other stages $i \in \{2, \dots, I\}$:

$$\begin{cases} T_i^c = \inf\{t \geq T_{i-1}^c \mid \int_{T_{i-1}^c}^{t} R_i(s)\,ds \geq \frac{m}{I}\}, \\ T_{i-1}^c = \sup\{t \leq T_i^c \mid \int_{t}^{T_i^c} R_i(s)\,ds \geq \frac{m}{I}\}, \end{cases} \tag{32}$$

and

$$C_i(t) = C_{i-1}(\sup\{t' \leq t \mid \int_{t'}^{t} R_i(s)\,ds \geq \frac{m}{I}\}). \tag{33}$$

Again, $\tilde{C}_i(t)$ is defined as in (33) but applied to $\tilde{C}_{i-1}(t)$.

The transfer phase of a peer lasts until the entire video file has been retrieved. Let T_I^c denote the time at which a peer that arrived at time T^a completes the final stage (and the whole transfer phase at the same time). There is no guarantee that the length of the transfer phase, $T_I^c - T^a$ (derived from (32)), is less than the length z of the playback phase with the full playback rate. If this is not the case, we assume that the playback is delayed so that the peer stays in the system until the end of the transfer phase and departs from the system just immediately thereafter. Thus, we have

$$\tilde{D}(t) = \min\{\tilde{A}(t-z), \tilde{C}_I(t)\}, \tag{34}$$

which completes our model.

VI. Steady State Synthesis

By extensively experimenting with the two models (fluid/stochastic), we have arrived at the following synthesis of the long-run behavior and performance of the segmented P2P VoD system.

(i) If

$$\frac{1}{\mu} < \frac{k}{\lambda}, \tag{35}$$

the number of leechers and non-permanent seeds are positive and finite and the equilibrium transfer rate for any leecher exceeds the playback rate w (so that the playback quality is sufficient for the VoD application). All stages are download constrained, and the numbers of leechers and non-permanent seeds are well estimated by equation (25).

(ii) If

$$\frac{k}{I\lambda} \leq \frac{1}{I\mu} < \frac{k}{\lambda}\left(1 - \frac{I-1}{Icz}\right), \tag{36}$$

the number of leechers and non-permanent seeds are positive and finite and the equilibrium transfer rate for any leecher exceeds the playback rate w (so that the playback quality is sufficient for the VoD application).

If additionally (24) is satisfied, then all stages are download constrained, and the numbers of leechers and non-permanent seeds are well estimated by equation (25). Otherwise, there is index $i^* \in \{0, 1, \dots, I-1\}$ such that the stages $1, \dots, i^*$ are download constrained and the rest $(i^* + 1, \dots, I)$ are upload constrained. In this case the numbers of leechers and non-permanent seeds are well estimated by Algorithm 1.

(iii) If

$$\frac{k}{\lambda}\left(1 - \frac{I-1}{Icz}\right) \leq \frac{1}{I\mu} < \frac{k}{\lambda}, \tag{37}$$

the number of leechers is still positive and finite but the number of non-permanent seeds becomes 0 and the equilibrium transfer rate for stage-I leechers stays below the playback rate w resulting in playback problems.

(iv) If

$$\frac{1}{I\mu} \geq \frac{k}{\lambda}, \tag{38}$$

the number of stage-I leechers goes to infinity (system becomes unstable), the number of non-permanent seeds becomes 0, and the equilibrium transfer rate for stage-I leechers stays below the playback rate w resulting in playback quality problems.

Note that cases (i)-(iv) divide the (λ, ζ)-plane into four complementary areas, each with its own characteristic behavior. The four areas are separated by three bordering lines determined by three threshold values for λ. Figure 1 illustrates this synthesis for $I = 3$ in the (λ, ζ)-plane.

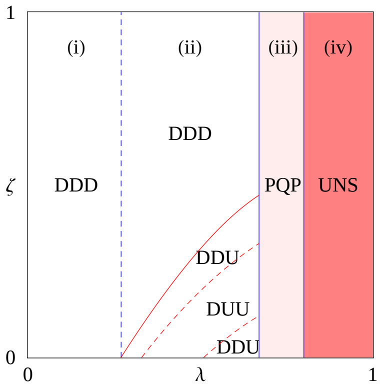

Fig. 1. Illustration of the steady-state synthesis for $I = 3$. The three vertical bordering lines satisfy (from left to right) $\lambda = k\mu$, $\lambda = Ik\mu\left(1 - \frac{I-1}{Icz}\right)$, and $\lambda = Ik\mu$, respectively. Letter D [U] refers to a download [upload] constrained stage (e.g., DDU = first two stages are D-constrained and last one is U-constrained), PQP refers to Playback Quality Problems, and UNS refers to an UNStable system with severe playback quality problems.

The altruism parameter ζ has no effect on the thresholds in equations (35), (36) and (37). However, the per-leecher service rate increases with ζ (up to the download capacity limitation d) implying that the system offers better viewing quality.

By (36), the limit for λ up to which the playback quality is sufficient increases with the number of permanent seeds, k, and the number of segments, I. Similarly, by (37), the limit for λ up to which the system remains stable increases (even linearly) with k and I.

Finally, we make a comment comparing the relative efficiency achieved with a segmented P2P approach compared to a traditional client-server approach used by typical content distribution services, such as YouTube and Akamai. In a client-server setting clients can download the file from k possible servers each with service rate μ. Then (35) corresponds to the *stability limit* of the system (limit for sufficient playback quality is even smaller). In the segmented p2p system, on the other hand, at this limit the system is able to offer the *best* possible quality (all stages are download constrained), and the actual limit where playback quality degrades below acceptable is even higher, recall (36) and the discussion above, than in the client-server setting.

VII. Numerical results and model validation

In this section, we validate the accuracy of the fluid model against (i) simulations from the stochastic model of Section V and (ii) traces from a more realistic BitTorrent simulator implementing a windowing algorithm, see [2]. To make the BitTorrent system behave in a segmented manner, the window is moved in steps so that each client must first download the entire segment at which point the window is moved to correspond to the next segment. Note that in the BitTorrent system the peers downloading the same segment are still able to share the data within their segment, while our models (fluid and stochastic) do not take this into account.

In the following tests we consider a viewing scenario where the parameters correspond to a typical YouTube setting. The users are viewing a video file consisting of 800 pieces each 32KB in size. The video coding rate is $w = 300$ kbit/s, and thus the viewing time is $z = 699$ s. The upload and download bandwidths of the users are $u = 512$ kbits/s and $d = 1024$ kbit/s (typical asymmetric ADSL subscriber rates). Also, we assume that $k = 10$ (ten permanent seeds). Similarly as in the steady-state analysis, we focus on the behavior of the system in the (λ, ζ) parameter space, cf., Figure 1.

A. Validation of system dynamics

First we consider the accuracy of the fluid model for predicting the dynamic properties of the stochastic model, as well as, the segmented BitTorrent system. In this setting, we assume the video is split into 10 segments, i.e., there are $I = 10$ stages, and new peers arrive with a rate $\lambda = 0.1$ 1/s. In the simulation of the stochastic model, customers arrived according to a Poisson process with rate λ and the results were obtained by averaging over 20 sample paths. In the BitTorrent traces, the arrivals were deterministic, but randomness is still caused by altruism, as modeled by ζ. The presented results represent one sample path of the system. The results are given in Figure 2. The figure shows the time evolution of the total number of leechers, $\sum_i x_i(t)$, and the number of seeds $y(t)$ as a function of time. In the figure, the result from the fluid model corresponds to the solid smooth lines, the stochastic model results are shown with dashed lines and the BitTorrent traces are depicted by the jagged solid lines.

The upper panel corresponds to the case with $\zeta = 0.7$. The system is download constrained and the number of seeds is higher than the number of leechers. In the lower panel, $\zeta = 0.3$ and the system is upload constrained (number of seeds has dropped below that of the leechers). From the results, we can observe that the dynamics are somewhat different between the fluid model and the stochastic model, while the match is better between the fluid model and the BitTorrent result. However, most importantly, the steady state solution of the fluid model clearly matches very well with the stochastic model and the BitTorrent. The accuracy of the fluid model is especially good in the download constrained case (upper panel).

The differences in the dynamics between the stochastic model and BitTorrent may be explained by the fact that the detailed mechanisms implemented in BitTorrent does not result in an equal sharing of the upload bandwidth, as is assumed in our stochastic model. This allows BitTorrent to more efficiently generate new seeds into the system, which significantly helps in shortening the time to reach steady state compared with the stochastic model. The fluid model also assumes the uniform sharing of the upload capacity. However, the utilized quasi-stationary approximation provides a counter-effect balancing the “slowdown” caused by the uniform sharing assumption.

B. Steady-state validation

Next we focus on validating the steady state synthesis results in Section VI. The synthesis results predict that the number of seeds stays positive as long as $\lambda < Ik\mu(1-(I-1)/(Icz))$, see (36). To study this, we have numerically evaluated the number of seeds in the system given by the steady state solution of the fluid model and we compare that result against simulated results from the stochastic model. The results are given in Figure 3, which depicts the number of non-permanent seeds, $\bar{y}$, as a function of λ for a system with $I = 5$ stages (upper panel) and $I = 10$ stages (lower panel). In each panel, we show the result for $\zeta = 0.7$ and $\zeta = 0.3$. In the figure, the solid lines without markers represent the solutions of the fluid model and the lines with circle markers represent the simulated results. The simulated results have been obtained by simulating the stochastic model for at least 60 000 s (in some cases up to 120 000 s) and discarding the results of the first 20 000 s due to initial transient. The final results are an average of 5 such sample paths. The threshold value for λ as predicted by (36) is indicated by the dashed vertical line.

The results demonstrate that indeed (nearly) up to the predicted threshold value of λ the number of seeds grows, and as λ is increased further beyond this the number of

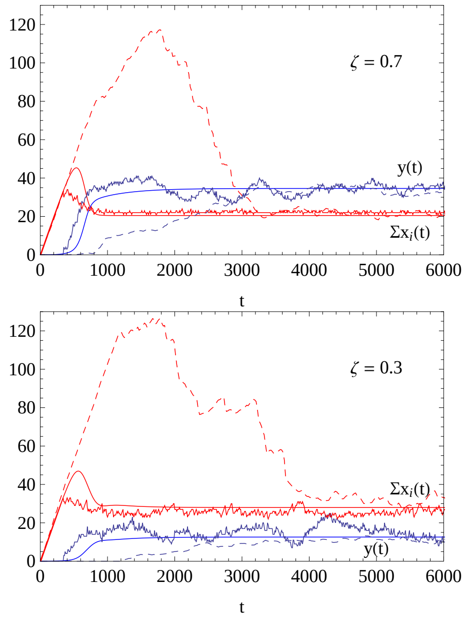

Fig. 2. Dynamic solution of $\sum_i x_i(t)$ and $y(t)$ as a function of t for the fluid model (solid smooth line), the stochastic model (dashed line) and the BitTorrent simulation (jagged solid line) with $\zeta = 0.7$ (upper panel) and $\zeta = 0.3$ (lower panel).

seeds quickly drops to zero indicating severe viewing quality problems. In the fluid model this change is sharp, while in the stochastic model it is more smooth. The accuracy of the fluid model in the case when $\zeta = 0.7$ is remarkably good (the system stays download constrained for all values of λ). However, when $\zeta = 0.3$ the system is download constrained for small values of λ but then switches to being upload constrained. In this case, the accuracy is not as good but still the qualitative behavior is accurately captured.

VIII. Conclusions

We have derived a fluid model for a segmented p2p system. Specific features of our model include an explicit modeling of the downloading efficiency via stages and the coupling of the actual video viewing time to the departure rate of the seeds. As demonstrated by our validation results with the stochastic model and the segmented BitTorrent system, the model accurately captures the steady state properties, while the dynamic behavior shows some differences.

Our main result concerns the steady-state properties of the system. In the (λ, ζ)-plane, we are able to characterize the performance of the system. Notably, an explicit condition has been derived under which the downloading rate exceeds the actual viewing rate, i.e., quality remains acceptable. The result allows us to characterize how the number of segments or the number of permanent seeds improves the system. Also, the result reveals that the condition is independent of altruism ζ. However, given that the threshold condition is satisfied, a higher level of altruism does imply better quality.

Future challenges include, analyzing the impacts of heterogeneous users. Also, generalizing the model to allow simultaneous downloading of multiple pieces could be useful for understanding the optimal allocation of uploading capacity between the segments.

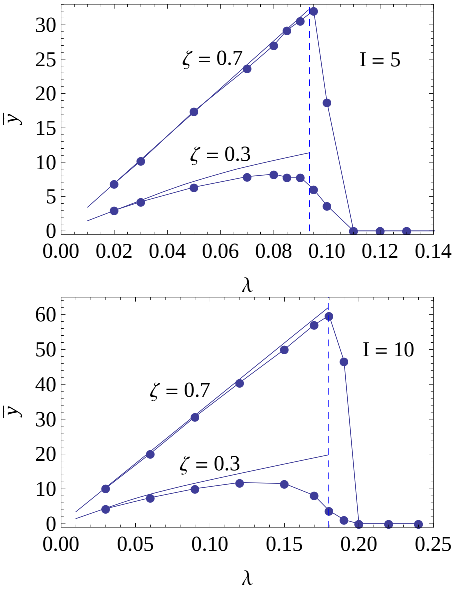

Fig. 3. Steady state solution of $\bar{y}$ as a function of λ when $\zeta = 0.3$ and $\zeta = 0.7$ with $I = 5$ (upper panel) and $I = 10$ (lower panel).

Acknowledgement

This work was supported by TEKES as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT).

References

[1] K. Parvez, C. Williamson, A. Mahanti, and N. Carlsson, "Analysis of BitTorrent-like protocols for on-demand stored media streaming," in *Proc. of ACM SIGMETRICS*, 2008, pp. 301–312.

[2] P. Savolainen, N. Raatikainen, and S. Tarkoma, "Windowing BitTorrent for video-on-demand: Not all is lost with tit-for-tat," in *Proc. of IEEE GLOBECOM*, 2008.

[3] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. Rodriguez, "Is high-quality vod feasible using P2P swarming?" in *Proc. of WWW*, 2007, pp. 903–912.

[4] M. Wang and B. Li, "R2: Random push with random network coding in live peer-to-peer streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1655–1666, December 2007.

[5] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: Optimal performance trade-offs," in *Proc. of ACM SIGMETRICS*, 2008, pp. 325–336.

[6] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for p2p streaming systems," in *Proc. of IEEE INFOCOM*, 2007, pp. 919–927.

[7] C. Feng and B. Li, "On large-scale peer-to-peer streaming systems with network coding," in *Proc. of ACM MM*, 2008, pp. 269–278.

[8] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proc. of ACM SIGCOMM*, 2004, pp. 367–378.

[9] D. Qiu and W. Sang, "Global stability of peer-to-peer file sharing systems," *Computer Communications*, vol. 31, no. 2, pp. 212–219, 2008.

[10] F. Lo Piccolo, G. Neglia, and G. Bianchi, "The effect of heterogeneous link capacities in BitTorrent-like file sharing systems," in *Proc. of HOT-P2P*, 2004, pp. 40–47.

[11] Y. Yue, C. Lin, and Z. Tan, "Analyzing the performance and fairness of BitTorrent-like networks using a general fluid model," *Computer Communications*, vol. 29, no. 18, pp. 3946–3956, 2006.