Analytical model for mesh-based P2PVoD

Lu, Y; Mol, JJD; Kuipers, FA; Van Mieghem, PFA

DOI
10.1109/ISM.2008.30

Publication date
2008

Document Version
Accepted author manuscript

Published in
Proceedings Tenth IEEE International Symposium on Multimedia

Analytical Model for Mesh-based P2PVoD

Yue Lu, Jan David Mol, Fernando Kuipers, Piet Van Mieghem
Delft University of Technology
{Y.Lu, J.J.D.Mol, F.A.Kuipers, P.F.A.VanMieghem}@tudelft.nl

Abstract

Recently, there has been a growing interest in academic and commercial environments for Video-on-Demand (VoD) using Peer-to-Peer (P2P) technology. Unlike centralized solutions for VoD services, P2P technology lets the clients distribute video content among themselves. In this paper, we propose an analytical model for P2PVoD and we compare that model to a realistic P2PVoD simulator. With our model, parameters that affect the system performance can be observed, and the system stability can be investigated. Our model leads to design rules for achieving a good and stable system performance. This work is, to our knowledge, the first analytical work to model mesh-based P2PVoD.

1 Introduction

Peer-to-Peer (P2P) applications are immensely popular on the Internet. Among these P2P applications, the current most popular application for file sharing is BitTorrent[1]. Besides file sharing, the mesh-based technology of BitTorrent is also deployed in P2P television (P2PTV) services like Coolstreaming [1] and PPLive [2]. In P2PTV users can access the available TV channels to view the content that is being displayed at that particular point in time. In P2P Video-on-Demand (P2PVoD, e.g. Tribler [3]) users arrive at arbitrary points in time into the system to watch a video of their choice from its beginning. The question addressed in this paper is *how to use mesh-based P2P technology to provide P2PVoD services with good and stable performance*. To answer this question, we have developed a P2PVoD model[2].

The rest of this paper is organized as follows: In Section 2, related work is discussed. In Section 3, we develop our analytical model for P2PVoD. This model aims to present the number of downloaders and seeds watching a video i and the average downloading speed at a peer as a function of time. After linearizing this analytical model in Section 4, we compare it with our simulation results in Section 5. Finally, we conclude in Section 6.

[1] http://www.bittorrent.com/

[2] In this paper when referring to P2PVoD, we mean mesh-based P2PVoD.

2 Related work

Guo *et al.* [4] and Qiu *et al.* [5] model mesh-based P2P file sharing analytically and present a performance study combined with extensive measurements.

Kumar *et al.* [6] proposed a fluid model for mesh-based P2PTV, which has only one seed. Lu *et al.* [7] proposed a mesh-based model for P2PTV and compared its blocking to that of IPTV. However, the models in [6] and [7] are not applicable to P2PVoD.

Some research works (e.g., [8], [9]) target mesh-based P2PVoD, but only use simulations to analyze which kind of chunk-scheduling method can achieve the best performance. The proposed system by Chi et al. [10] was evaluated with the help of analytical models, but what they analyzed is tree-based P2PVoD [11], not mesh-based P2PVoD.

Prior to this work, an analytical model of mesh-based P2PVoD seemed missing.

3 A general fluid model for P2PVoD

Before modeling and analyzing P2PVoD, its basic mechanism and characteristics are addressed in order to better understand the behavior of the P2PVoD system. The content of P2PVoD is a video lasting a fixed amount of time. The video in P2PVoD can be divided into chunks. Each video has a unique ID, e.g. i. All chunks of video i can be found at the seeds. The distribution of video i starts with one initial seed, the original video i content provider.

A peer arrives at arbitrary points in time into the system to watch video i from its beginning. In our model, each P2PVoD peer stores the video i's content on his computer until he stops viewing this video. Thus, once a peer obtains a chunk, he makes the chunk available for downloading by other peers until he leaves.

We refer to the peer who is still downloading as "downloader" and refer to the peer who already finished the download, but is still viewing the video as "seed." A peer joins the system as a downloader and contacts other peers[3] in order to download chunks of video i. After a prebuffering period, the peer starts the playback and from then on the video content is displayed, while at the same time the near-future video content is downloaded. After the peer has finished downloading the whole video file, he will become a seed until he departs.

In our P2PVoD model, there are two kinds of peer departures. One of them is the random departure and the other is the definite departure. A downloader may leave the network randomly at rate θ_i before the download is completed (e.g., when he feels that the video is boring). Even though a peer becomes a seed, he may still be viewing the video. Hence, a seed may leave the network randomly at a rate of θ_i before the video playback has completed. Nevertheless, the seed will definitely leave after he has viewed the video, with definite seed leaving rate $\gamma_i(t)$ (we can consider $1/\gamma_i(t)$ to be the seed serving time, see Fig. 1).

A peer generally obtains video chunks in playback order. Hence, a peer has to download the near-future video chunks as high-priority within a downloading time limit.

In our P2PVoD network, beside seeds who will definitely upload data, a downloader who has not finished downloading yet can also upload data to other downloaders. The number of downloaders at t is $x_i(t)$ and the number of seeds at t is $y_i(t)$. A downloader has probability $\eta_i(t)$ to be used for sharing his content with others. Based on our simulations, the value of $\eta_i(t)$ is approaching 1 except for the first few seconds of the system. Hence, we can simplify the model by setting $\eta_i(t) = 1$.

We list the symbols, which will be used in our P2PVoD model, in Table 1.

Table 1. Symbols

v: Video playback rate (Mbit/s).
L_i: The length (in seconds) of video i.
$\lambda_i(t)$: Peers' arrival rate for video i at time t.
θ_i: Peer's random leaving rate from video i.
$\gamma_i(t)$: Seed's definite leaving rate. $\gamma_i(t) = \frac{1}{\text{seed serving time}}$.
$x_i(t)$: No. of downloaders in the video i system at t.
$y_i(t)$: No. of seeds in the video i system at t.
bw_{os}: The upload rate of the original source provider.
bw_{up}: Avg. upload rate at a peer for video delivery.
bw_{down}: Max download rate of a peer for video delivery.
$u_i(t)$: Avg. download rate of a peer at t in video i system.
$T_i(t)$: Time a peer needs for downloading the video i at t.
τ_i: The time interval from the time that video i appeared to the time that the first seed appears in the system.

[3] A new peer will choose some other peers who are also watching this video to form a neighbor group. Within this group, he can download what he needs from other peers based on the chunk availability information.

3.1 Model description

Our analysis of mesh-based P2PVoD can be considered to be a worst-case study. We do not consider any extra complex strategies (like peer selection, incentive management, failure management, chunk scheduling, etc.). We use a fluid model to compute the time-dependent average number of downloaders and seeds in system i. Hence, we do not compute any fluctuations around the average.

We will show that our general P2PVoD model leads to a non-linear system. Consequently, we shall analyze which factors cause this non-linearity such that we might redesign our P2PVoD system to become linear in all conditions.

In the following, we introduce ordinary differential equations to express our fluid model in general.

The total uploading rate of the system can be expressed as $\min\{bw_{down}x_i(t), bw_{up}(x_i(t) + y_i(t)) + bw_{os}\}$, where bw_{down} is the download rate upperbound for video delivery; bw_{up} is the average upload rate at a peer for video delivery; bw_{os} is the upload rate of the original source provider (we assume only one original source provider for one video); $x_i(t)$ and $y_i(t)$ respectively represent the number of downloaders and seeds for video i at time t. If there is enough downloading bandwidth, the total uploading rate of the system reduces to $bw_{up}(x_i(t) + y_i(t)) + bw_{os}$. At time t, the overall downloading rate related to video i is equal to the overall uploading rate related to video i: $u_i(t)x_i(t) = \min\{bw_{down}x_i(t), bw_{up}(x_i(t)+y_i(t))+bw_{os}\}$. Hence, we express the average download rate $u_i(t)$ as

$$u_i(t) = \frac{\min\{bw_{down}x_i(t), bw_{up}(x_i(t) + y_i(t)) + bw_{os}\}}{x_i(t)} \tag{1}$$

In order to analyze the system performance for video i, we need to calculate $u_i(t)$. In order to obtain $u_i(t)$, we should first get the values of $x_i(t)$ and $y_i(t)$, which can be obtained by solving Eqs. (2) to (5), explained below.

Each peer joins the P2PVoD system as a downloader. After finishing the download, a downloader will become a seed. At time t, the total downloading rate $u_i(t)x_i(t)$ (Mbit/s) divided by the length of the video $L_i v$ (Mbits) can be considered as the rate at which downloaders become seeds. Continuing with this idea, the downloaders' generating rate $\frac{dx_i(t)}{dt}$ should be equal to the downloaders' arrival rate $\lambda_i(t)$ minus the downloaders' leaving rate $\theta_i x_i(t)$ and minus the rate of downloaders becoming seeds $\frac{u_i(t)}{L_i v}x_i(t)$.

$$\frac{dx_i(t)}{dt} = \lambda_i(t) - \theta_i x_i(t) - \frac{\min\{bw_{down}x_i(t), bw_{up}(x_i(t)+y_i(t))+bw_{os}\}}{L_i v} \tag{2}$$

In our fluid model, the peer arrival process can be any kind of process. For simplicity, we consider the average arrival rate as a constant value, $\lambda_i(t) = \lambda_i$.

The seeds' generating rate $\frac{dy_i(t)}{dt}$ should be equal to the rate of downloaders becoming seeds $\frac{u_i(t)}{L_i v} x_i(t)$ minus the seeds' leaving rate $(\theta_i + \gamma_i(t))y_i(t)$. Thus,

$$\begin{aligned} \frac{dy_i(t)}{dt} \quad &= \quad \frac{\min\{bw_{down}x_i(t), bw_{up}(x_i(t)+y_i(t))+bw_{os}\}}{L_i v} \\ &\quad -(\theta_i + \gamma_i(t))y_i(t), \end{aligned} \tag{3}$$

Eqs. (2) and (3) show that there still is one unknown variable: the seed definite departure rate $\gamma_i(t)$. We deduce $\gamma_i(t)$ below.

If we make a peer (seed) depart as soon as the display ends[4], our P2PVoD system is possibly non-linear, where the seed's definite departure rate $\gamma_i(t)$ depends on $x_i(t)$ and $y_i(t)$.

We obtained the equations of $\gamma_i(t)$ and $T_i(t)$ based on Fig. 1. When a peer finishes downloading the video, it will become a seed until the video finishes and the peer departs. If a peer downloads the data very fast (high downloading speed), this peer will have a longer seed service time. The service time of a seed at time t, regardless of the peer arbitrary departures during the viewing, is equal to $1/\gamma_i(t) = L_i + B_u - T_i(t)$. For a given peer, the downloading time $T_i(t)$ times the downloading rate $u_i(t)$ is equal to the video size $L_i v$.

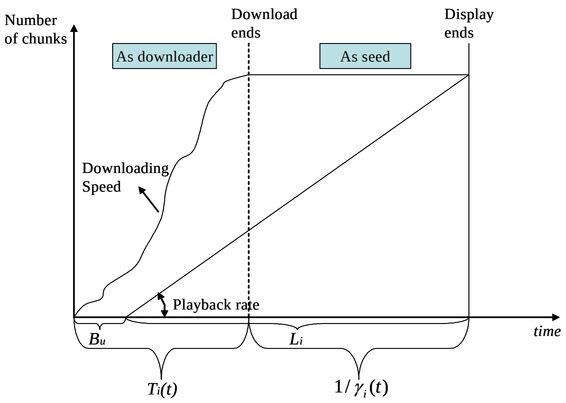

Figure 1. A peer U changes from a downloader status to a seed status.

When assuming that a peer definitely departs as soon as the display ends, we obtain

$$\gamma_i(t) = \frac{1}{L_i + B_u - T_i(t)}, \tag{4}$$

[4]The seed serving time depends on the average download time, which is determined by the number of downloaders and seeds in the system.

where the downloading time equals

$$T_i(t) = \max\left\{\frac{L_i v}{bw_{down}}, \frac{x_i(t)L_i v}{bw_{up}(x_i(t)+y_i(t))+bw_{os}}\right\}, \tag{5}$$

Eqs. (4) and (5) indicate that $T_i(t)$ may depend on $x_i(t)$ and $y_i(t)$, while $\gamma_i(t)$ depends on $T_i(t)$; thus $\gamma_i(t)$ depends on $x_i(t)$ and $y_i(t)$. Eq. (3) shows that $y_i(t)$ depends on $\gamma_i(t)$. Thus, when the download capacity is large, the seed definite leaving rate $\gamma_i(t)$ depends on $x_i(t)$ and $y_i(t)$, making $x_i(t)$ and $y_i(t)$ non-linear.

After having introduced the general idea of how to use differential equations to model a P2PVoD system, we are going to analyze four phases of the system: Start-up phase, Seed Appearance (SA) phase, Seed Departure (SD) phase, and Steady-state.

3.2 Start-up phase $(0 \le t < \tau_i)$

The system starts with one original source provider. Thus, the initial number of downloaders is $x_i(0) = 0$. We use $y_i(t)$ to express the number of seeds in the system at time t. Here, $y_i(t)$ excludes the original source provider and $y_i(t) = 0$ when $0 \le t < \tau_i$. The number of seeds in the system stays zero until a peer finishes downloading the whole video file and becomes the first seed. At the very beginning phase of the video i system, $0 \le t < \tau_i$, the first downloader is able to download video content with the download rate of $u_i(t)$ and $\tau_i = L_i v / u_i(t)$.

We define the Start-up phase as the time interval between the availability of the video and the appearance of the first seed:

$$0 \le t < \tau_i \begin{cases} \frac{dx_i(t)}{dt} = \lambda_i - \theta_i x_i(t), \\ y_i(t) = 0, \\ u_i(t) = \frac{bw_{up}x_i(t)+bw_{os}}{x_i(t)} \approx bw_{up} \end{cases}$$

3.3 Seed Appearance phase $(\tau_i \le t < L_i)$

We assume that all peers are able to finish the download before the display ends. We define the SA phase as the time interval between the appearance of the first seed and the definite departure of the first seed. In this phase, no definite departures of seeds occur and no videos are released.

Thus, Eqs. (2) and (3) are simplified with $\gamma_i(t)$=0.

3.4 Seed Departure phase $(t \ge L_i)$

After the SA phase, the system enters the so-called SD phase ($t \ge L_i$). We define the SD phase as the time interval between the departure of the first seed to the start of steady-state.

In this phase, the seed's definite leaving rate $\gamma_i(t)$ is not equal to zero anymore. Eqs. (4) and (5) show that the value of $\gamma_i(t)$ depends on $T_i(t)$, which has different expressions under different conditions and which depends on $x_i(t)$ and $y_i(t)$ when $\frac{L_i v}{bw_{down}} < \frac{x_i(t)L_i v}{bw_{up}(x_i(t)+y_i(t))+bw_{os}}$.

The conditions referred to above will be analyzed in Section 3.5 on the steady-state. However, the conditions deduced for the steady-state can also be used to analyze the SD phase, when we consider the SD phase (varying with time t) as built-up by many quasi-steady states, each of which is lasting a unit (e.g., 1 second) of time. We can analyze the SD phase at different time points varying around different equilibrium points $\{\bar{x}_i, \bar{y}_i\}$ deduced in Section 3.5 with different values of γ_i.

In Section 5.1, we have used Matlab to compute the results of $x_i(t)$ and $y_i(t)$ based on the Eqs. (2) to (5) and the various conditions presented in the following section. The conditions relate to the value of γ_i and the condition that either the upload bandwidth or the download bandwidth is the constraint.

3.5 Steady-state

Analogous to the steady-state analysis for P2P file sharing in [5], we can find expressions for our P2PVoD model.

In steady-state, nothing varies with time t. Hence, Eqs. (2) and (3) become

$$\lambda_i - \theta_i\bar{x}_i - \min\{\frac{bw_{down}\bar{x}_i}{L_i v}, \frac{bw_{up}(\bar{x}_i+\bar{y}_i)+bw_{os}}{L_i v}\} = 0$$

$$\min\{\frac{bw_{down}\bar{x}_i}{L_i v}, \frac{bw_{up}(\bar{x}_i+\bar{y}_i)+bw_{os}}{L_i v}\} - (\theta_i+\gamma_i)\bar{y}_i = 0$$

where $\bar{x}_i = \lim_{t\to\infty} x_i(t)$ and $\bar{y}_i = \lim_{t\to\infty} y_i(t)$ are the equilibrium values of $x_i(t)$ and $y_i(t)$.

1) We can solve these equations if $\frac{bw_{down}\bar{x}_i}{L_i v} \leq \frac{bw_{up}(\bar{x}_i+\bar{y}_i)+bw_{os}}{L_i v}$ (the download bandwidth is the constraint) as

$$\bar{x}_i = \frac{\lambda_i}{\frac{bw_{down}}{L_i v} + \theta_i} \tag{6}$$

$$\bar{y}_i = \frac{\lambda_i}{(\gamma_i+\theta_i)(1+\frac{\theta_i L_i v}{bw_{down}})} \tag{7}$$

where $\gamma_i = \frac{1}{L_i + B_u - (L_i v / bw_{down})}$.

With the expressions of $\bar{x}_i$ and $\bar{y}_i$, the assumption that $\frac{bw_{down}\bar{x}_i}{L_i v} \leq \frac{bw_{up}(\bar{x}_i+\bar{y}_i)+bw_{os}}{L_i v}$ amounts to

$$\frac{L_i v}{bw_{down}} \geq \frac{\lambda_i L_i v - (\frac{\lambda_i bw_{up}}{\gamma_i+\theta_i} + bw_{os})}{\lambda_i bw_{up} + \theta_i bw_{os}}$$

Thus, when $\lambda_i L_i v > \frac{\lambda_i bw_{up}}{\gamma_i+\theta_i} + bw_{os}$ (which is equivalent to $\gamma_i > \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$), if $\frac{L_i v}{bw_{down}} \geq \frac{\lambda_i L_i v - (\frac{\lambda_i bw_{up}}{\gamma_i+\theta_i}+bw_{os})}{\lambda_i bw_{up}+\theta_i bw_{os}}$, we use (6) and (7) to express the number of downloaders and seeds in steady-state. When $\lambda_i L_i v \leq \frac{\lambda_i bw_{up}}{\gamma_i+\theta_i} + bw_{os}$ (which is equivalent to $\gamma_i \leq \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$), $\frac{L_i v}{bw_{down}} > 0$ will be always larger than $\frac{\lambda_i L_i v - (\frac{\lambda_i bw_{up}}{\gamma_i+\theta_i}+bw_{os})}{\lambda_i bw_{up}+\theta_i bw_{os}}$ even if the download bandwidth is *not* the constraint, then we can also use (6) and (7).

2) On the other hand, if $\frac{bw_{down}\bar{x}_i}{L_i v} > \frac{bw_{up}(\bar{x}_i+\bar{y}_i)+bw_{os}}{L_i v}$ (the upload bandwidth is the constraint), we obtain

$$\bar{x}_i = \frac{\lambda_i\theta_i L_i v + \lambda_i\gamma_i L_i v - bw_{up}\lambda_i - bw_{os}\theta_i - bw_{os}\gamma_i}{S} \tag{8}$$

$$\bar{y}_i = \frac{\lambda_i bw_{up} + bw_{os}\theta_i}{S} \tag{9}$$

where $S = bw_{up}\gamma_i + \theta_i\gamma_i L_i v + \theta_i^2 L_i v$, $\gamma_i = \frac{1}{L_i + B_u - (L_i v/u_i)}$ and $u_i = \frac{bw_{up}(\bar{x}_i+\bar{y}_i)+bw_{os}}{\bar{x}_i}$.

With the expressions of $\bar{x}_i$ and $\bar{y}_i$ above, the assumption that $\frac{bw_{down}\bar{x}_i}{L_i v} > \frac{bw_{up}(\bar{x}_i+\bar{y}_i)+bw_{os}}{L_i v}$ can also be expressed as

$$0 < \frac{L_i v}{bw_{down}} < \frac{\lambda_i L_i v - (\frac{\lambda_i bw_{up}}{\gamma_i+\theta_i} + bw_{os})}{\lambda_i bw_{up} + \theta_i bw_{os}}$$

Only when $\lambda_i L_i v > \frac{\lambda_i bw_{up}}{\gamma_i+\theta_i} + bw_{os}$, and if $\frac{L_i v}{bw_{down}} < \frac{\lambda_i L_i v - (\frac{\lambda_i bw_{up}}{\gamma_i+\theta_i}+bw_{os})}{\lambda_i bw_{up}+\theta_i bw_{os}}$, Eqs. (8) and (9) should be used.

This analysis shows that the characteristics of the seed's definite departure rate $\gamma_i(t)$ directly determine whether the system equations are linear or not. The downloading time of the whole file $T_i(t) = \frac{L_i v}{bw_{down}}$ in (5) leads to $\gamma_i(t)$, which is independent of $x_i(t)$ and $y_i(t)$, resulting in linear system equations; while $T_i(t) = \frac{x_i(t)L_i v}{bw_{up}(x_i(t)+y_i(t))+bw_{os}}$ in (5) leads to non-linear system equations. In the following section, we will redesign our P2PVoD model, such that it becomes linear in all conditions. If the conditions (e.g., the bandwidth of end users or the user behavior) cannot be controlled, it is important to linearize the model to achieve a stable system performance *in all conditions*.

For the Start-up phase and Seed Appearance phase, there is no effect of $\gamma_i(t)$, because $\gamma_i(t) = 0$. These two phases always lead to linear equations. However, the Seed Departure phase and the steady-state need linearization, because $\gamma_i(t)$ may depend on $x_i(t)$ and $y_i(t)$ in these two phases.

4 Linearization of the P2PVoD model

In order to obtain a linear system under all conditions, the seed serving time $\frac{1}{\gamma_i(t)}$ must be constant (i.e. a peer departs a fixed amount of time after his download finishes[5]).

[5]It will be no problem for a P2PVoD developer to achieve this (just make the video content stored at a seed stop being shared, even when he is still viewing the video).

We can design P2PVoD applications to obey this rule. The linear differential equations can be expressed as

$$\frac{dZ(t)}{dt} = A_j Z(t) + b_j, \quad j = 1, 2 \tag{10}$$

where $Z(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix}$.

There are two possibilities ($j = 1$ and $j = 2$) based on the conditions deduced in Section 3.5:

1. Case $j = 1$, where $\gamma_i \leq \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$:

   Whether the system reaches a steady-state is only determined by bw_{down}, even when the download bandwidth is large:

   $A_1 = \begin{bmatrix} -(\theta_i + \frac{bw_{down}}{L_i v}) & 0 \\ \frac{bw_{down}}{L_i v} & -(\theta_i + \gamma_i) \end{bmatrix}$ and

   $b_1 = \begin{bmatrix} \lambda_i \\ 0 \end{bmatrix}$

   The eigenvalues [12] of A_1 are $\mu_1 = -(\theta_i + \frac{bw_{down}}{L_i v})$ and $\mu_2 = -(\theta_i + \gamma_i)$, which are both negative.

   Since both eigenvalues are negative, we have a stable system that converges exponentially fast in t to the steady-state.

   (a) Steady-state ($t \to \infty$):
   $\begin{bmatrix} \bar{x}_i \\ \bar{y}_i \end{bmatrix} = \frac{\lambda_i}{(\theta_i + \frac{bw_{down}}{L_i v})(\theta_i + \gamma_i)} \begin{bmatrix} (\theta_i + \gamma_i) \\ \frac{bw_{down}}{L_i v} \end{bmatrix}$,
   which gives the same[6] expressions as (6) and (7).

   (b) Sensitivity of eigenvalues:
   A larger value of the normalized download upperbound rate $\frac{bw_{down}}{L_i v}$ and a larger value of γ_i, under the condition that $\gamma_i \leq \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$, will cause the eigenvalues to have larger negative values, which on its turn will cause the number of downloaders and seeds $x_i(t)$ and $y_i(t)$ to reach a steady-state faster.

2. Case $j = 2$, where $\gamma_i > \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$:

   $A_2 = \begin{bmatrix} -(\theta_i + \frac{bw_{up}}{L_i v}) & -\frac{bw_{up}}{L_i v} \\ \frac{bw_{up}}{L_i v} & \frac{bw_{up}}{L_i v} - (\theta_i + \gamma_i) \end{bmatrix}$ and

   $b_2 = \begin{bmatrix} \lambda_i - \frac{bw_{os}}{L_i v} \\ \frac{bw_{os}}{L_i v} \end{bmatrix}$

   The eigenvalues of A_2 are

   $\mu_1 = \frac{-(2\theta_i + \gamma_i) + \sqrt{\gamma_i^2 - 4\frac{bw_{up}}{L_i v}\gamma_i}}{2}$ and $\mu_2 = \frac{-(2\theta_i + \gamma_i) - \sqrt{\gamma_i^2 - 4\frac{bw_{up}}{L_i v}\gamma_i}}{2}$. Although complex, both eigenvalues always have negative real parts, which again shows that the system is stable.

   (a) Steady-state ($t \to \infty$):
   $\begin{bmatrix} \bar{x}_i \\ \bar{y}_i \end{bmatrix} = \frac{1}{(\theta_i + \frac{bw_{up}}{L_i v})(\theta_i + \gamma_i - \frac{bw_{up}}{L_i v}) + (\frac{bw_{up}}{L_i v})^2}$
   $\begin{bmatrix} (\theta_i + \gamma_i - \frac{bw_{up}}{L_i v})(\lambda_i - \frac{bw_{os}}{L_i v}) - \frac{bw_{up} bw_{os}}{(L_i v)^2} \\ \frac{bw_{up}}{L_i v}(\lambda_i - \frac{bw_{os}}{L_i v}) + \frac{bw_{os}}{L_i v}(\theta_i + \frac{bw_{up}}{L_i v}) \end{bmatrix}$,
   which is the same as Eqs. (8) and (9).

   (b) Sensitivity of eigenvalues:
   Given $\gamma_i > \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$, a larger value of γ_i will make the system reach the steady-state faster.

   On the other hand, if the download bandwidth is the constraint, the model equation changes to be the same as for case 1.

[6] Under the condition that γ_i is independent of $\bar{x}_i$ and $\bar{y}_i$.

Based on these formulae deduced above, we can examine the effect of some parameters on the system behavior:

1) *What is the effect of* bw_{os}*?* The upload bandwidth of the original source provider bw_{os} does not affect the steady-state at all when $\gamma_i \leq \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$ or when the download bandwidth is the constraint. When $\gamma_i > \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$ and the upload bandwidth is the constraint, the larger the bw_{os}, the less downloaders and the more seeds in steady-state, which leads to a larger average download rate $u_i(t)$ according to Eq. (1). Hence, a larger bw_{os} is helpful for the system performance when the average seed serving time is small and the peers' upload bandwidth is limited.

2) *What is the effect of* bw_{up}*?* Assuming all peers are able to finish the download before the display ends, if we change the average upload bandwidth of the normal peers bw_{up} when $\gamma_i \leq \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$ or when the download bandwidth is the constraint, the number of downloaders and seeds in steady-state will not change. When $\gamma_i > \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$ and the upload bandwidth is the constraint, the larger the bw_{up}, the less downloaders and the more seeds in steady-state if $\gamma_i < \lambda_i$. Hence, a larger bw_{up} is helpful for the system performance when the average seed serving time is small and the peers' upload bandwidth is limited.

3) *What is the effect of the peer arrival rate* λ_i*?* Assuming $\theta_i = 0$ and ignoring the comparably small $\frac{bw_{os}}{L_i v}$, if we double the peer arrival rate λ_i, we can find that the number of downloaders and seeds in steady-state will be doubled in all conditions. Hence, we can probably normalize our system equations in steady-state, with the number of peers divided by λ_i.

4) *What is the effect of the seed serving time* $1/\gamma_i$*?* If we consider the two conditions individually, a larger value of the seed definite departure rate γ_i will make the system reach the steady-state faster. However, if we consider the two conditions simultaneously, our best choice

is to set the value of γ_i close to, but not exceeding, $\frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$. As such we will have a stable system that not only converges faster to the steady-state, but also contains a larger number of seeds. For further analyzing the effect of this factor, detailed experiment results will be shown in Section 5.2.

5 Experiments

In this section, we compare our analytical results with our simulation results, under the same conditions.

For the computational part, we can feed the parameters into our fluid model and solve the ordinary differential equations with Matlab. We set the parameters as shown in Table 2.

Table 2. Value of parameters in experiments

$v = 0.5Mbit/s$ (for a video with TV quality),
$bw_{up} = 0.9Mbit/s$,
$bw_{os} = 4Mbit/s$,
$bw_{down} = 10Mbit/s$,
$\lambda_i = 1$, $B_u = 10\,\mathrm{sec}$,
$L_i = 5\,\mathrm{min}$ (e.g., a short YouTube-like video clip).

For the simulation part, we have set up a discrete-event simulator. The transmission of chunks in the simulator is discrete, as opposed to our fluid model. A policy is therefore needed to determine for each peer which chunk it will download from which neighbor. The chunks are transferred from peer i to a neighbor j as follows. Peer i keeps its neighbors informed about the chunks it has finished downloading. Peer j can request these chunks and each request is granted by appending the chunk to the send buffer of peer i. To increase the chunk availability, while maintaining VoD behavior, we let peers download chunks at random within a window of B_u seconds starting from the first chunk that has not yet been downloaded. Since playback starts B_u seconds after the download starts, a peer typically notices no difference due to this change in policy.

Each simulation starts with one initial seed, and the peers arrive according to a Poisson process. The different departure processes will be explained in Section 5.2 and 5.1 individually. The simulation results are averaged over 20 runs. We found the average bandwidth utilization rate of a peer (upload rate/upload capacity) to be equal to 80% on average, while the bandwidth utilization rate of the original source provider is nearly 1. Hence, in order to be consistent with the settings in our fluid model, we set the original source provider's upload capacity to $4Mbit/s$, and a normal peer's upload capacity to $\frac{0.9}{80\%} = 1.1Mbit/s$. The other parameters are equal to the fluid model.

Figs. 2, 3 and 4 show the number of downloaders and seeds, as well as the average download rate as a function of time t. We can imagine that the more popular video i is, the more downloaders and seeds there will be in the system.

5.1 General non-linear system

Peers arrive at a rate of λ_i and depart only when the playback is finished ($\theta_i = 0$). Each peer stores the video i's content until this video ends displaying ($\gamma_i(t)$ depends on $x_i(t)$ and $y_i(t)$).

Fig. 2 illustrates that, since the video was made available, the number of seeds increased until it reached a steady-state, while the number of downloaders increased at first and then decreased suddenly into a steady-state. The average downloading rate at a peer is small in the start-up phase and reaches its maximum in the steady-state.

Comparing the analytical results and simulation results, they closely match except for the time of the start-up phase and SA phase. That is because we let the average download rate in start-up phase $u_i(0 \leq t \leq \tau_i)$ be roughly equal to bw_{up} in our mathematical model, while in the real case (in simulation), $u_i(0 \leq t \leq \tau_i) = bw_{up} + \frac{bw_{os}}{\lambda_i(t)t} > bw_{up}$, which leads to a smaller $\tau_i = L_i v / u_i(t)$. Correspondingly, the peak number of downloaders in the start-up phase in the simulation was a little bit smaller than the one in the mathematical model. Nevertheless, the peak number of downloaders and seeds in our simulations, as well as the steady-states, still closely match the analytical results. Our fluid model can thus be used to predict these numbers, which can then be used in the design of P2PVoD algorithms. Until seeds actually depart (before SD phase), the number of seeds plus the number of downloaders can be derived from the arrival rate and has to be equal in both the analysis and simulations. Then, in the SA phase, the differences between the results for the seeds and the downloaders are equal, which can be observed in Fig. 2.

The differences in download speed are due to the same reason. Our simulation tracks the average download speed with a history of 10 seconds, and combined with a constant arrival of peers with an initial download speed of 0. Thus, the average download speed is lower in the simulation than is predicted by the fluid model.

Other differences between the analytical and simulation results can be caused by the possible peer correlation and the flexible and stochastic P2P network under simulation. This includes slower, but more fluent, transitions between states when compared to the fluid model.

With our settings above, this non-linear system seems to perform very well. However, non-linear systems might be unstable. For instance, if too many ADSL peers are watching this video i (if we change $bw_{up} = 0.9Mbit/s$ to $bw_{up} = 0.4Mbit/s$), on average there will be no peers able to finish the download before the display ends. In this case, there will be no seeds and the average download rate will be always smaller than the playback rate, which causes blocking everywhere in this P2PVoD system.

5.2 Linearized system

We use the same values for our parameters, except for the value of γ_i. Because $\frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} \approx 0.00616$, we set:

(1) $\gamma_i = 0.006$ (which can be considered the threshold); Each user keeps his stored video for $\frac{1}{0.006} \approx 167$ seconds after he finishes the download, no matter how fast he downloaded it (see Fig. 3).

We can see that this linear case with $\gamma_i < \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}}$ has a similar system performance as its non-linear counterpart, because the average download rate at a peer is similar, even though the number of seeds in this case is smaller in the steady-state. The difference between the analytical results and the simulation results are similar to those in the non-linear system.

Furthermore, once $\gamma_i < \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}}$, the average download rate will always be maximized. Therefore keeping the video at least $\frac{\lambda_i L_i v - bw_{os}}{\lambda_i bw_{up}}$ seconds after becoming a seed is indeed helpful to lead to a better system performance. Thus, the threshold of $\frac{\lambda_i L_i v - bw_{os}}{\lambda_i bw_{up}}$ is meaningful for P2PVoD application developers.

(2) $\gamma_i = 0.008$; Each user keeps his stored video for $\frac{1}{0.008} = 125$ seconds after he finishes the download. The situation of the linear system in this case is shown in Fig. 4.

In this case with $\gamma_i > \frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}}$, not only the number of seeds, but also the average download rate is smaller than the cases before, which leads to worse system performance.

We can observe from both analytical results and simulation results that the average download rate is not stable any more, but decreases sharply at around 400 seconds; even though it rebounds after that, it has a smaller value than in the previous cases. In Fig. 4, because the peers cannot download at full speed after 400 seconds in simulations, the peer correlation becomes critical for the downloaders to obtain the desired chunks. As a result, the differences between the fluid model and the discrete simulations increase. Nevertheless, the trends of the simulation results and the analytical results remain similar.

6 Conclusion

In this paper, we have modeled mesh-based P2PVoD. This model, which is based on current P2PVoD applications, leads to non-linear system equations. In a non-linear model, small perturbations of the input may lead to large (undesirable) changes in the behavior of the system. Consequently, we have provided rules for a P2PVoD application that ascertain a linear behavior. A critical factor is the seed definite leaving rate γ_i. The best choice for P2PVoD application developers is to set it close to, but not exceeding, the value of $\frac{\lambda_i bw_{up}}{\lambda_i L_i v - bw_{os}} - \theta_i$.

With our model, parameters that affect the system performance were observed and analyzed in this paper. The results from realistic simulations match well with our analytical model. Our model can thus be used to predict the system behavior, which can aid in the design of P2PVoD systems.

References

[1] X. Zhang, J. Liu, B. Li, and TS. P. Yum, "CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media streaming," Proc. of IEEE INFOCOM, vol. 3, pp. 2102-2111, March 2005.

[2] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A Measurement Study of a large-Scale P2P IPTV System," IEEE Transactions on Multimedia, vol. 9, no. 8, December 2007.

[3] J.A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D.H.J. Epema, M. Reinders, M. van Steen, H.J. Sips, "Tribler: A social based Peer to Peer system," Proc. of IPTPS, Feb. 27-28, 2006.

[4] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "A Performance Study of BitTorrent-like Peer-to-Peer Systems," IEEE Journal on selected areas in communications, vol. 25, no. 1, January 2007.

[5] D. Qiu and S. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer networks," Proc. of ACM SIGCOMM 2004, August 2004.

[6] R. Kumar, Y. Liu, and K. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," Proc. of IEEE INFOCOM 2007.

[7] Y. Lu, F.A. Kuipers, M. Janic, and P. Van Mieghem, "E2E blocking probability of IPTV and P2PTV," Proc. of IFIP Networking 2008, May 2008.

[8] A. Vlavianos, M. Iliofotou, and M. Faloutsos, "BiToS: Enhancing BitTorrent for supporting Streaming Applications," Proc. of IEEE INFOCOM 2006.

[9] J.J.D. Mol, J.A. Pouwelse, M. Meulpolder, D.H.J. Epema, and H.J. Sips, "Give-to-Get: Free-riding-resilient Video-on-Demand in P2P Systems", Proc. of SPIE, MMCN 2008.

[10] H. Chi, Q. Zhang, J. Jia and X. Shen, "Efficient Search and Scheduling in P2P-based Media-on-Demand Streaming Service", IEEE Journal on selected areas in communications, VOL.25, NO.1, January 2007.

[11] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," in Journal of Peer-to-Peer Networking and Applications, by Springer New York, Feburary, 2008.

[12] G. Arfken, "Eigenvectors, Eigenvalues.", in Mathematical Methods for Physicists, 3rd ed. Orlando, FL: Academic Press, pp. 229-237, 1985.

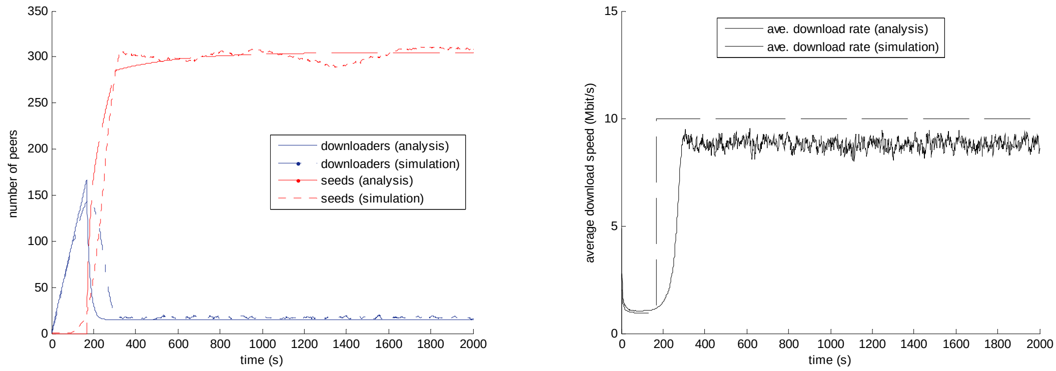

Figure 2. The number of downloaders $x_i(t)$ and seeds $y_i(t)$ (left) and the average download speed per peer $u_i(t)$ (right) as a function of time in a non-linear video i system.

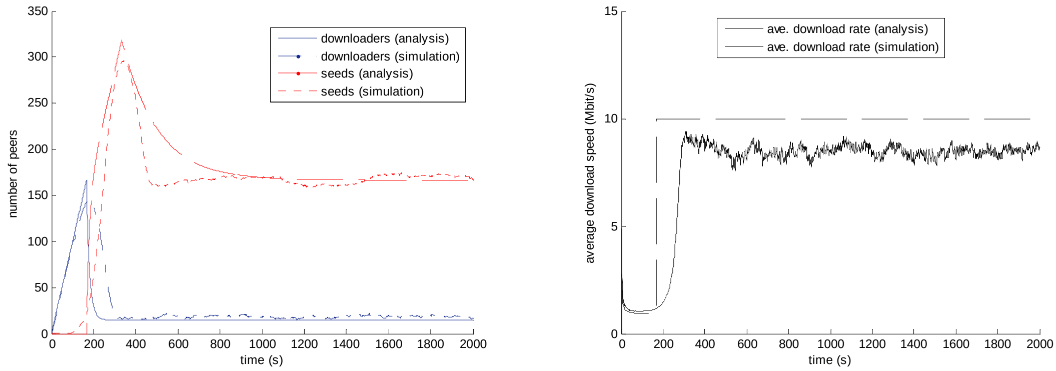

Figure 3. The number of downloaders $x_i(t)$ and seeds $y_i(t)$ (left) and the average download speed per peer $u_i(t)$ (right) as a function of time in a linearized system with γ_i=0.006.

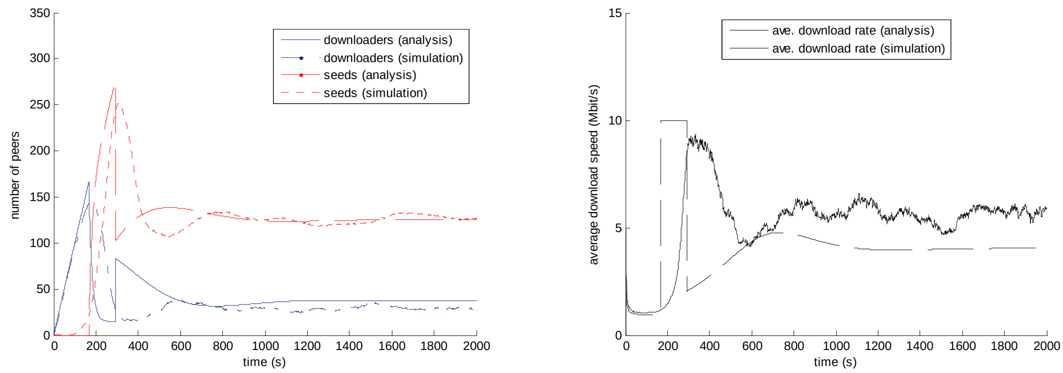

Figure 4. The number of downloaders $x_i(t)$ and seeds $y_i(t)$ (left) and the average download speed per peer $u_i(t)$ (right) as a function of time in a linearized system with γ_i=0.008.