

# Stability and Capacity of Peer-to-Peer Assisted Video-on-Demand Applications

Franco Robledo Amoza\*  
frobledo@fing.edu.uy

Pablo Rodríguez-Bocca\*  
prbocca@fing.edu.uy

Pablo Romero\*  
promero@fing.edu.uy

Claudia Rostagnol\*  
crostag@fing.edu.uy

\* Departamento de Investigación Operativa, Facultad de Ingeniería, Universidad de la República.  
Julio Herrera y Reissig 565, 11300, Montevideo, Uruguay.

**Abstract**—We propose a mathematical framework for the optimal design of a video on-demand (VoD) application under regime. Peers join the network following a poissonian process, download progressively one or possibly many concurrent video contents, and abort the system when they wish. The system is supported by static servers managed by the operator, called *super-peers*, and the altruism of peers, that upload resources and might stay online even after completing downloading.

Our goal is to minimize the expected download time perceived by end-users under regime. We propose a general fluid model, and show that it is stable, reaching its regime. Via the Little's law, we find closed expressions for the expected waiting time under regime in terms of the popularity of different contents, file sharing efficiency and other network parameters. We state theoretically that this system outperforms traditional Content Delivery Networks (CDN). The operator can decide only the number of video replicas stored in each super-peer, a fact that has a direct impact on the mean waiting times. Hence, a combinatorial optimization problem is introduced, whose nature is similar to the multi-knapsack problem (i.e. the items are video contents, and the knapsacks are the super-peers storage). A greedy randomized resolution is here designed, and a comparison between traditional content distribution systems promotes the deployment of peer-to-peer video on-demand services.

Finally, real-life scenarios are studied based on traces taken from YouTube. The results confirm that the peer-to-peer model can outperform five times the throughput of traditional CDN under flash-crowded scenarios.

**Index Terms**—Peer-to-Peer, Video on-demand, Fluid Model, Combinatorial Optimization Problem.

## I. INTRODUCTION

Peer-to-peer (P2P) networks are self-organized communities virtually installed over the Internet infrastructure, in which their participants, called peers, both receive and share their resources (CPU-time, bandwidth and/or contents). The P2P philosophy represents an attractive alternative to traditional Content Delivery Networks (CDN), statically deployed with multiple servers. The cooperation between peers in a P2P system provides high scalability properties and, at the same time, low operational costs. Nowadays, P2P computing is a dynamic research area, and mathematical tools play a fundamental role in their understanding and design, in part because an experimental fail in an on-line test will disappoint

users, with serious drawbacks. Peer evolution and scalability, service capacity, file sharing efficiency and free riding are major causes of concern during the mathematical analysis of these networks.

There are basically three video streaming modes. The simplest one is file-sharing, where the video content is first generated, then distributed and downloaded by peers, and finally watched just after completing the download. The second is on-demand streaming, where the video is distributed as progressive download to end-users. In progressive download protocols, the video is downloaded on the end-user's device in a best effort mode, because the protocols do not include any consumer-producer synchronization (i.e. real-time constraints), and user can start watching before completing the download. The third mode is called live-streaming, where the video content is generated, distributed and played by peers at the same time with strict real-time requirements. In this case, a streaming protocol is used, and there is an explicit synchronization between consumers and producer.

A very inspiring system for replication and fast dissemination of files is BitTorrent, created by Bram Cohen [1], [2]. BitTorrent was originally designed for file sharing applications. However, nowadays most of the P2P applications over the Internet are BitTorrent-based. One of such applications is the GoalBit Video Platform [3], [4], currently used to share live and on-demand video streaming over a P2P network. In these systems, peers are either *downloaders* when they actively download content; or *seeders*, once they finished the download process but remain connected, sharing the already downloaded contents. There is also an entity or node named *tracker*, which knows all the peers that are seeding or downloading a content. GoalBit introduces a third type of node to the network named *super-peer*. These kind of nodes have higher bandwidth resources than a normal peer, and usually join the network with longer life-times (very stable peers). The role of super-peer is to store and forward the contents to common peers (with a very short life in the system). In the current GoalBit protocol specification, super-peers are nodes managed by the operator of the platform, hosted in the cloud, and implement a specific caching policy [3].

In this paper we propose a general fluid model to address peer evolution and measure expected download times under regime. The main purpose is to decide the number of video

replicas that must be stored in each super-peer, in order to improve the quality of experience of end-users.

This paper is structured as follows. Section II contains a summary of related work. Section III introduces a general fluid model, in which peers can concurrently download several contents, and are classified according to the number of simultaneous downloads. Two special cases are discussed in-depth in Section IV. The first is a concurrent model for BitTorrent-based networks. The second derived from the most general is a sequential fluid model, in which each peer downloads video contents sequentially. We find closed expressions for the expected waiting times in both models, regarding P2P and CDN systems separately, and prove that the performance of a P2P system is never worse than that of a CDN. A combinatorial optimization problem (COP) is introduced and solved in Section V, trying to allocate video replicas in super-peers in order to minimize the expected waiting time. Given that the COP has similarities with the Multi-knapsack Problem (which is Strongly NP-Hard) we develop a greedy randomized heuristic. Real-life scenarios based on YouTube traces are analyzed in Section VI. Finally, Section VII contains the main conclusions and an brief enumeration of several aspects for future research.

## II. RELATED WORK

Usually, the development of experiments in real P2P systems is expensive, and a fail has a disappointing effect in final users. As a consequence, the scientific community works to develop mathematical models in order to predict the behavior of the system. In [5], Yang and de Veciana justify mathematically the consistency of the service capacity of P2P file sharing services. They propose a branching process, and state that a P2P system highly outperforms a traditional CDN. A basic Markovian model is also introduced to describe peer evolution. In [6], Qiu and Srikant analyze BitTorrent-like systems under steady state and its variability, showing empirical validation as well. A steady state analysis is first presented with a simple fluid model, in which the peer evolution is captured by exogenous poissonian arrivals and exponential departures in the system. They consider homogeneous peers, and find a closed expression for the expected waiting time. A sensitivity analysis of this performance measure with respect to different design parameters offers one of the first insights of the BitTorrent's soundness. A special treatment is included for the file sharing efficiency between peers, and states the robustness of random peer selection. The steady state is partially characterized as locally stable, and it was conjectured that it is globally stable as well. The conjecture is true, and proved for the first time in [7]. This work is generalized in a first stage in [8], extending the model for several concurrent multi-torrents. The authors argue that most BitTorrent users download several files at the same time. A second generalization can be found in [9], where the authors introduce the presence of super-peers in the network, and study the steady state of a video on-demand application. Here, we propose a further generalization of the mathematical approach considered in [9]. A general

fluid model is presented, adding node churn, valuable network parameters and the fact that peers (as well as seeders) may abort the system when they wish, even before downloading the complete video file. This general model can be used as an analytical framework for future research.

We study concurrent and sequential on-demand video streaming, in which the swarm is assisted by super-peers, managed by the operator. Our goal is to distribute video contents over the GoalBit platform, planning the efficient usage of storage capacity. The scheduling must attend videos' popularity, and looks forward to minimize the mean expected download time for end-users. We include a stability analysis of the fluid model, and prove that the P2P network consistently outperforms traditional CDN.

## III. GENERAL FLUID MODEL

Consider an open network which offers  $K$  video contents with sizes  $\{s_1, \dots, s_K\}$  measured in Megabits. Peers join the network, download progressively one or possibly many concurrent video contents and abort the system when they wish. Peers are then classified in exhaustive and mutually disjoint sets  $C^1, \dots, C^K$ , where  $C^i$  is the set of peers that download  $i$  video contents simultaneously. Denote  $x_j^i(t)$  the number of peers from class  $C^i$  that are downloading video  $j$  in a certain instant  $t$ . They join the network following a poissonian process of respective rates  $\lambda_j^i$ , and abort the system with exponential law, and respective rates  $\theta_j^i$ . The number of seeders owning exactly  $i$  contents, and seeding video  $j$  at instant  $t$  are denoted by  $y_j^i(t)$ , and depart the system exponentially with rates  $\gamma_j^i$ . We shall assume identical peers, with respective upload and download capacities denoted by  $\mu$  and  $c$  (measured in Mbps). Peers also contribute with the system uploading video contents although they do not have the entire file as seeders. The file sharing efficiency between peers is a coefficient  $\eta : 0 \leq \eta \leq 1$  that indicates the sharing percentage between downloaders. All seeders from class  $C^i$  that own video  $j$  can decide a portion  $\alpha_j^i$  of their available uploading capacity, in order to feed downloaders of video  $j$ . Super-peers behave like seeders, but they do not leave the system. They are denoted by  $z_j^i$ , and have upload capacity  $\rho$ . All this information can be summarized in a general fluid model (GFM), specified as follows:

$$\frac{dx_j^i}{dt} = \lambda_j^i - \theta_j^i x_j^i(t) - \min\{c_j^i x_j^i(t), \eta \mu_j^i x_j^i(t) + \sum_k (\mu_j^k y_j^k(t) + \rho_j^k z_j^k)\}$$
(1)

$$\frac{dy_j^i}{dt} = \min\{c_j^i x_j^i(t), \eta \mu_j^i x_j^i(t) + \sum_k (\mu_j^k y_j^k(t) + \rho_j^k z_j^k)\} - \gamma_j^i y_j^i(t),$$
(2)

where additionally:

- 1)  $\mu_j^i = \frac{\mu_i}{s_j}$  is the class  $C^i$  upload rate for video  $j$ , and  $\sum_k \mu_k = \mu$ .
- 2)  $c_j^i = \frac{c_i}{s_j}$  is the class  $C^i$  download rate for video  $j$ , and  $\sum_k c_k = c$ .
- 3)  $\rho_j^i = \frac{\rho_i}{s_j}$  is the class  $C^i$  upload rate of video  $j$  for super-peers, and  $\sum_k \rho_k = \rho$ .

- 4)  $\theta_j^i$  is the departure rate of peers.
- 5)  $\gamma_j^i$  is the departure rate of seeders.

The minimum function in the second side of the equalities means that the bottleneck is either in downloading or uploading. The GFM is a non-linear switched system of ordinary differential equations. We will denote for short  $[n] = \{1, \dots, n\}$ .

#### IV. TWO OUTSTANDING SUB-MODELS

##### A. Concurrent Fluid Model (CFM)

The number of variables involved in the GFM force us to assume further hypothesis in order to analyze the stationary state of the system and have an insight of the super-peers optimal behavior, which are the only nodes that can be managed by the network operator. Inspired in BitTorrent-based systems, we will strictly stick to the following assumptions:

- 1) “Fair transmission”: the resources are equally distributed in the different concurrent videos:  $\mu_i = \frac{\mu}{i}$ ,  $c_i = \frac{c}{i}$  and  $\rho_i = \frac{\rho}{i}$ .
- 2) “Tit-for-tat”: Peers in class  $C^i$  that at time  $t$  are downloading video  $j$  receive from all other downloaders a streaming rate proportional to the upload bandwidth  $\mu_j^i$  and their population:

$$\left( \frac{\mu_j^i x_j^i(t)}{\sum_k \mu_j^k x_j^k(t)} \right) \sum_k \eta \mu_j^k x_j^k(t) = \eta \mu_j^i x_j^i(t).$$

- 3) “Fair Seeders”: Peers from class  $C^i$  that at time  $t$  are downloading video  $j$  receive from all the seeders a streaming rate proportional to the download bandwidth and their population:

$$\left( \frac{c_j^i x_j^i(t)}{\sum_k c_j^k x_j^k(t)} \right) \sum_k \mu_j^k y_j^k(t) = \alpha_j^i \sum_k \mu_j^k y_j^k(t).$$

- 4) “Fair Super-peers”: Analogously, peers from class  $C^i$  that at time  $t$  are downloading video  $j$  receive from all the super-peers a streaming rate proportional to the download bandwidth and their population:  $\alpha_j^i \sum_k \rho_j^k z_j^k$ .
- 5) “Peers Departures”: the peers and seeders departures follow the *Zipf law*, being linearly decreasing with respect to the numbers of concurrent video downloads:  $\gamma_j^i = \gamma/i$ , and  $\theta_j^i = \theta/i$ .

After including these BitTorrent-based assumptions to the GFM we get the P2P Concurrent Fluid Model (P2P-CFM):

$$\frac{dx_j^i}{dt} = \lambda_j^i - \frac{\theta}{i} x_j^i - \min \left\{ \frac{c}{is_j} x_j^i, \eta \frac{\mu}{is_j} x_j^i + \alpha_j^i \sum_k \left( \frac{\mu}{s_j} \frac{y_j^k}{k} + \frac{\rho}{s_j} \frac{z_j^k}{k} \right) \right\} \quad (3)$$

$$\frac{dy_j^i}{dt} = \min \left\{ \frac{c}{is_j} x_j^i, \eta \frac{\mu}{is_j} x_j^i + \alpha_j^i \sum_k \left( \frac{\mu}{s_j} \frac{y_j^k}{k} + \frac{\rho}{s_j} \frac{z_j^k}{k} \right) \right\} - \gamma_j^i y_j^i, \quad (4)$$

where the independent variable  $t$  is omitted for short. Table 1 summarizes the symbolic notation.

$K$	available videos
$s_j$	size of video-item $j$
$x_j^i(t)$	downloaders in class $C^i$ downloading video $j$ at time $t$
$y_j^i(t)$	seeders in class $C^i$ seeding video $j$ at time $t$
$z_j^i(t)$	super-peers in class $C^i$ seeding video $j$ at time $t$
$\lambda_j^i$	arrival rate for peers in class $C^i$ requesting video $j$
$\theta_j^i$	departure rate of peers in class $C^i$ requesting video $j$
$\gamma_j^i$	departure rate of seeders in class $C^i$ seeding video $j$
$c$	total download bandwidth for each peer.
$\mu$	total upload bandwidth for each peer.
$\rho$	total upload bandwidth for each super-peer.
$\eta$	exchange efficiency between peers ( $\eta \in [0, 1]$ ).

Fig. 1. Symbology of the Concurrent Fluid Model.

##### B. Rest Point Analysis for CFM

If we find a time  $t$  such that simultaneously  $\frac{dx_j^i(t)}{dt} = \frac{dy_j^i(t)}{dt} = 0$  for every pair  $i, j \in [K]$ , the system will rest indefinitely in the same constant vector state  $(\bar{x}_j^i, \bar{y}_j^i)$ . This is called a stationary state. Now, we will find the stationary state for the P2P-CFM.

**Proposition IV.1.** *The rest point for the P2P-CFM is:*

$$\bar{x}_j^i = \max \left\{ \frac{i \lambda_j^i s_j}{\theta s_j + c}, \frac{i \lambda_j^i (\lambda_j - \bar{\rho}_j - \bar{\phi}_j)}{\lambda_j (\theta + \frac{\eta \mu}{s_j} - \frac{\mu \theta}{\gamma s_j})} \right\} \quad (5)$$

$$\bar{y}_j^i = \frac{\lambda_j^i - \theta_j^i \bar{x}_j^i}{\gamma_j^i}, \quad (6)$$

where  $\bar{\rho}_j = \sum_k \rho_j^k z_j^k$  and  $\bar{\phi}_j = \frac{\mu \lambda_j}{\gamma s_j}$ .

*Proof:* Summing Equations (3) and (4) we immediately prove Expression (6). When the download capacity is the system’s bottleneck then the minimum function is equal to  $\frac{c}{is_j} x_j^i$ , so the steady state for the P2P-CFM can be found solving a linear system of equations:

$$\bar{x}_j^i = \frac{i \lambda_j^i s_j}{\theta s_j + c} \quad (7)$$

Denote for short  $\bar{\rho}_j = \sum_k \rho_j^k z_j^k$  and  $\bar{\phi}_j = \sum_k \frac{\mu_j^k \lambda_j^k}{\gamma_j^k} = \frac{\mu \lambda_j}{\gamma s_j}$ .

On the other hand, when the upload capacity is the system’s bottleneck, summing (3) for all  $i \in [K]$  we get:

$$\sum_i \frac{\bar{x}_j^i}{i} = \frac{\left( \theta + \frac{\eta \mu}{s_j} - \frac{\mu \theta}{\gamma s_j} \right)}{\lambda_j - \bar{\phi}_j - \bar{\rho}_j}. \quad (8)$$

Additionally, Equation (3) can be re-written:

$$\lambda_j^i = \left( \theta + \frac{\eta \mu}{s_j} - \frac{\mu \theta}{\gamma s_j} + \frac{\bar{\rho}_j + \bar{\phi}_j}{\sum_i \frac{\bar{x}_j^i}{i}} \right) \frac{\bar{x}_j^i}{i} \quad (9)$$

Replacing (8) into (9) and spreading the expression, we obtain the desired result. ■

A traditional CDN (with a client-server paradigm) can be viewed as a particular case of this analytical approach.

Specifically, users do not cooperate ( $\mu = 0$ ), seeders do not participate in the network ( $y_j^i(t) = 0$ ) and the previously named super-peers are now static servers. Replacing these parameters in Expression (3), the CDN Concurrent Fluid Model (CDN-CFM) is defined by the following system of ordinary differential equations:

$$\frac{dx_j^i(t)}{dt} = \lambda_j^i - \theta \frac{x_j^i}{i} - \min \left\{ \frac{c}{s_j} \frac{x_j^i(t)}{i}, \alpha_j^i(t) \sum_k \rho_j^k z_j^k \right\} \quad (10)$$

The expressions for the steady state in the CDN can be immediately obtained making  $\mu = 0$  in Equation (5):

$$\bar{x}_{jCDN}^i = \max \left\{ \frac{i\lambda_j^i s_j}{\theta s_j + c}, \frac{i\lambda_j^i (\lambda_j - \bar{\rho}_j)}{\theta \lambda_j} \right\} \quad (11)$$

Now we can compare the capacity of the P2P-CFM and CDN-CFM systems. Let us assume stability for a moment (we prove asymptotic stability of a special but important case in the following subsection). Denote  $T_{CFM}^{P2P}$  and  $T_{CFM}^{CDN}$  the expected waiting times under regime for the respective systems *P2P-CFM* and *CDN-CFM*. The following proposition is intuitive, and sounds:

**Proposition IV.2.**  $T_{CFM}^{P2P} \leq T_{CFM}^{CDN}$

*Proof:* Consider the random variable  $T_j^i$  that represents the waiting time for user-type  $(j, i)$  (a member of class  $x_j^i$ ). By the Little's law we relate the mean waiting time with the number of users under regime:

$$E(T_j^i) = \frac{\bar{x}_j^i}{\lambda_j^i}. \quad (12)$$

Equality (12) holds for both systems (P2P and CDN), where the number of users are  $\bar{x}_{jP2P}^i$  and  $\bar{x}_{jCDN}^i$  respectively. Let us denote  $X$  to the random variable that represents the class of certain arrival in the GFM. It has range  $R_X = \{(j, i) : j, i \in [K]\}$ , and  $P(X = (j, i))$  is the probability that a certain new arrival is from class  $x_j^i$ . The poissonian arrivals with intensity rates  $\lambda_j^i$  imply that  $P(X = (j, i)) = \frac{\lambda_j^i}{\lambda}$ , being  $\lambda$  the global sum rate. The mean waiting time of a user in the P2P GFM can be found via conditional expectation:

$$\begin{aligned} E(T) &= E(E(T/X)) = \\ &= \sum_i \sum_j P(X = (j, i)) E(T/X = (j, i)) \\ &= \sum_i \sum_j \frac{\lambda_j^i}{\lambda} E(T_j^i) \\ &= \frac{1}{\lambda} \sum_i \sum_j \lambda_j^i \bar{x}_j^i, \end{aligned}$$

where we used the Little's law. Notice that equalities hold again for both systems:

$$\begin{aligned} T_{CFM}^{P2P} &= \frac{1}{\lambda} \sum_i \sum_j \lambda_j^i \bar{x}_{jP2P}^i, \\ T_{CFM}^{CDN} &= \frac{1}{\lambda} \sum_i \sum_j \lambda_j^i \bar{x}_{jCDN}^i. \end{aligned}$$

Hence, it suffices to prove that the number of downloaders in the *P2P* fluid model is never greater than the one in the CDN model:  $\bar{x}_{jP2P}^i \leq \bar{x}_{jCDN}^i$ . We use Expressions (11), (5) and elementary algebra. If the download is the system's bottleneck then the equality is obvious. Otherwise, the second argument of the maximum function in Expression (5) must dominate, and moreover its numerator  $i\lambda_j^i(\lambda_j - \bar{\rho}_j - \bar{\phi}_j)$  is positive. The following chain of inequalities holds:

$$\begin{aligned} \bar{x}_{jP2P}^i &= \frac{i\lambda_j^i(\lambda_j - \bar{\rho}_j) - i\lambda_j^i \bar{\phi}_j}{\lambda_j \left( \theta + \frac{\mu}{s_j} \left( \eta - \frac{\theta}{\gamma} \right) \right)} \\ &< \frac{i\lambda_j^i(\lambda_j - \bar{\rho}_j)}{\theta \lambda_j} \\ &= \bar{x}_{jCDN}^i. \end{aligned}$$

This inequality holds if and only if

$$(-i\lambda_j^i \bar{\phi}_j)(\theta \lambda_j) < i\lambda_j^i(\lambda_j - \bar{\rho}_j) \left( \lambda_j \frac{\mu}{s_j} \left( \eta - \frac{\theta}{\gamma} \right) \right).$$

Using that  $\bar{\phi}_j = \frac{\mu \lambda_j}{\gamma s_j}$  and canceling common factors, the inequality holds if and only if

$$\lambda_j \frac{\theta}{\gamma} + (\lambda_j - \bar{\rho}_j) \left( \eta - \frac{\theta}{\gamma} \right) > 0.$$

Equivalently, if and only if  $\eta \lambda_j > \bar{\rho}_j \left( \eta - \frac{\theta}{\gamma} \right)$ . But the latter inequality is obviously true, since  $\lambda_j \geq \bar{\rho}_j + \bar{\phi}_j > \bar{\rho}_j$ . ■

The remaining of this paper focuses on an outstanding case, defined by a single-class system, where each peer downloads exactly one content at a time.

### C. Sequential Fluid Model (SFM)

In this subsection we will study the GFM in the particular case in which peers download exactly one video content at a time (the single-class case -  $i = 1$ ). We will call it P2P-Sequential Fluid Model (P2P-SFM):

$$\begin{aligned} \frac{dx_j}{dt} &= \lambda_j - \theta_j x_j(t) - \min\{c_j x_j(t), \eta \mu_j x_j(t) + \mu_j y_j(t) + \rho_j z_j\} \\ \frac{dy_j}{dt} &= \min\{c_j x_j(t), \eta \mu_j x_j(t) + \mu_j y_j(t) + \rho_j z_j\} - \gamma y_j(t). \end{aligned}$$

A direct calculation shows that:

$$\begin{aligned} \bar{y}_{jP2P}^{SFM} &= \frac{\lambda_j - \theta \bar{x}_{jP2P}^{SFM}}{\gamma_j} \\ \bar{x}_{jP2P}^{SFM} &= \max \left\{ \frac{\lambda_j s_j}{\theta s_j + c}, \frac{\lambda_j (\gamma s_j - \mu) - \gamma \rho z_j}{\theta (\gamma s_j - \mu) + \eta \gamma \mu} \right\}. \end{aligned}$$

P2P-SFM is a linear-switched system, i.e. a special class of dynamic system. The global stability of a particular case has been studied in [7]. There, the global stability is proved, but the authors do not address any performance analysis. Recall that an equilibrium point  $\bar{x}$  of a dynamic system is stable if there exists a positive radius  $R$  such that for any arbitrary  $\bar{R} < R$  we can find  $r < \bar{R}$  that the orbit  $x(t)$  is inside the ball  $B(r, \bar{x})$  whenever the initial point is inside it. Additionally,  $\bar{x}$  is

asymptotically stable if it is stable and there is a radius  $\bar{R} > 0$  such that the orbit  $x(t)$  converges to  $\bar{x}$  whenever  $x(0) \in B(\bar{x}, \bar{R})$ . The reader can find these and further definitions in [10].

**Theorem IV.3.** *The SFM is globally stable whenever  $\gamma_j > 0$  for all  $j \in [K]$ .*

*Proof:* We will sketch the main idea of the proof. The SFM consists of  $K$  independent systems of two linear-switched ordinary differential equations. Without loss of generality, we can study one of those blocks (thus omitting subscripts):

$$\begin{aligned} \frac{dx}{dt} &= \lambda - \theta x(t) - \min\{cx(t), \eta\mu x(t) + \mu y(t) + \rho z\} \\ \frac{dy}{dt} &= \min\{cx(t), \eta\mu x(t) + \mu y(t) + \rho z\} - \gamma y(t). \end{aligned}$$

The main idea is to prove that the evolution  $(x(t), y(t))$  stays forever after a finite time either in Area I (where the upload is the bottleneck) or Area II, which are disjoint and mutually exhaustive cases. In both cases, the local stability of linear differential equations suffices, and is proved via elementary linear algebra. ■

The reader can write-out a complete proof from [7], where the authors study a very similar system, but with a single file and no super-peer assistance (i.e. without the constant term  $\rho z$  in the minimum argument). It has some tricks when the starting point is in the border of both areas. Indeed, in [7] the authors prove global stability of a linear switched system, which has many similarities with the SFM. For instance, if the peer evolution  $x_j(t)$  rests eventually in a linear zone, Proposition IV.3 will assure global stability for the SFM.

We briefly review the traditional CDN for this important single-class system (CDN Sequential Fluid Model, or CDNSFM):

$$\frac{dx_j}{dt} = \lambda_j - \theta_j x_j(t) - \rho_j z_j$$

The CDNSFM is globally stable, and the peer population converges to the rest point  $\bar{x}_j$ :

$$\bar{x}_{jCDN}^{SFM} = \frac{\lambda_j - \rho_j z_j}{\theta_j}$$

In this case the complete evolution for the peer population can be found:  $x_j(t) = \bar{x}_{jCDN}^{SFM}(1 - e^{-\theta t}) + x_j(0)$ .

#### D. Expected Waiting Times

The performance of the Peer-to-Peer Video on-demand sequential system is never worse than its equivalent CDN version:

**Theorem IV.4.**  $T_{P2P}^{SFM} \leq T_{CDN}^{SFM}$

*Proof:* Again, the equality holds when the download is the system's bottleneck. Otherwise, we will show that  $\bar{x}_{jP2P}^{SFM} < \bar{x}_{jCDN}^{SFM}$ , and the result follows from the Little's law.

We will construct an auxiliary inequality to conclude linearly the proof. Observe that  $\eta\mu\lambda_j > 0 > -\rho_j z_j \theta$ . Adding the term  $\theta s_j \lambda_j$  we have:

$$(\theta s_j + \eta\mu)\lambda_j > (\lambda_j s_j - \rho_j z_j)\theta.$$

Multiplying by the negative factor  $-\gamma\mu$  on both sides:  $-\gamma(\theta s_j + \eta\mu)\lambda_j \mu < -\gamma(\lambda_j s_j - \rho_j z_j)\mu\theta$ . Now, we add the positive term  $(\gamma\theta s_j + \eta\mu\gamma)(\gamma\lambda_j s_j - \gamma\rho_j z_j)$  on both sides, to get:

$$\begin{aligned} &(\gamma\lambda_j s_j - \gamma\rho_j z_j)(\gamma\theta s_j + \eta\mu\gamma - \mu\theta) > \\ &(\gamma\theta s_j + \eta\mu\gamma)(\gamma\lambda_j s_j - \gamma\rho_j z_j - \lambda_j \mu) \end{aligned}$$

We can rewrite the last inequality as follows:

$$\frac{\gamma\lambda_j s_j - \gamma\rho_j z_j}{\gamma\theta s_j + \eta\mu\gamma} > \frac{\lambda_j(\gamma s_j - \mu) - \gamma\rho_j z_j}{\gamma\theta s_j - \mu\theta + \eta\mu\gamma}$$

and the proof follows linearly:

$$\begin{aligned} \bar{x}_{jP2P}^{SFM} &= \frac{\lambda_j(\gamma s_j - \mu) - \gamma_j \rho_j z_j}{\theta \gamma s_j - \mu\theta + \eta\mu\gamma} \\ &< \frac{\gamma\lambda_j s_j - \gamma\rho_j z_j}{\gamma\theta s_j + \eta\mu\gamma} \\ &= \frac{\lambda_j s_j - \rho_j z_j}{\theta s_j + \eta\mu} \\ &< \frac{\lambda_j s_j - \rho_j z_j}{\theta s_j} = \bar{x}_{jCDN}^{SFM}. \end{aligned}$$

Theorem IV.3 is implicitly used during the proof, that assumes the system converges to the rest point, i.e.  $x_j(t)_{P2P}^{SFM} \rightarrow \bar{x}_{jP2P}^{SFM}$ , and  $x_j(t)_{CDN}^{SFM} \rightarrow \bar{x}_{jCDN}^{SFM}$ . In order to understand the consistency of the obtained results we will analyze the sensibility of the expected time  $T_j^{SFM}$  for video  $j$ , with respect to the network parameters: entry rates, abortion rates, file sharing efficiency, sizes of the different video contents and super-peers capacities. We would like to firstly remark that the number of peers under the rest point does not depend on the seeders aborting rate if it is large enough. In fact, when  $\gamma s_j \gg \mu$  we have that

$$\bar{x}_j^{SFM} \approx \frac{\gamma(\lambda_j s_j - \rho_j z_j)}{\gamma(\theta s_j + \eta\mu)} = \frac{\lambda_j s_j - \rho_j z_j}{\theta s_j + \eta\mu}. \quad (13)$$

In real-life networks, the seeders usually abort immediately after completing the download, and Expression (13) is indeed a good approximation. By Theorem IV.3, the SFM converges to the rest point. Via the Little's law we can find a rough approximation for the expected download times for video  $j \in [K]$  in the SFM:

$$T_j^{SFM} = \frac{\bar{x}_{jP2P}^{SFM}}{\lambda_j} = \frac{1}{\lambda_j} \frac{\lambda_j s_j - \rho_j z_j}{\theta s_j + \eta\mu} \quad (14)$$

By direct derivation of Expression (14) with respect to the network parameters, it can be observed that:

- 1) The waiting times are monotonically increasing with respect to the sizes of the contents. This is consistent with our intuitive idea that bigger files will take more time to be downloaded.
- 2) If the entry rates increase, peers will wait more. This is a common element of waiting systems with limited resources.

- 3) When the abortion rates of peers  $\theta$  increases, the expected waiting times are consequently reduced. It is evident that peers that depart before downloading will experiment lower time excursions, whereas the number of peers under steady state is hence reduced. The departure rates play the role of a *decay* in the entry rate.
- 4) Naturally, when the sharing efficiency  $\eta$  is increased, the throughput of the system increases as well, and the mean waiting times are consequently reduced.
- 5) Finally, the throughput of the system increases with the super-peers capacity  $\rho$ , and the number of replicas for video  $j$  in the network,  $z_j$ .

## V. COMBINATORIAL OPTIMIZATION PROBLEM

### A. Description

This section has the main problem of this paper. The goal is to minimize the mean waiting times in a progressive video on-demand system, assisted by super-peers managed by the operator. For the sake of simplicity and efficiency, we will focus on the P2P sequential fluid model (P2P-SFM), in which closed forms can be obtained for the mean waiting time for each video content.

Let us denote  $X$  to the random variable that represents the class of an entry peer in the SFM. The poissonian process with intensity rates  $\lambda_i$  imply that  $P(X = i) = \frac{\lambda_i}{\lambda}$ , being  $\lambda$  the sum rate. The mean waiting time of a user in the P2P-SFM can be found analogously to the GFM (via the Little's law and conditional expectation):

$$E(T) = \frac{1}{\lambda} \sum_{j=1}^K \bar{x}_{jP2P}^{SFM}$$

Accordingly, the mean waiting time is proportional to the whole population size, so we will minimize the latter.

We must decide the number of video replicas among  $P$  super-peers. The decision variable is a binary matrix  $E$  of size  $P \times K$ , whose entries are  $E(p, j) = 1$  if and only if we store video content  $j \in [K]$  in super-peer  $p \in [P]$ . We also impose that every video content must be duplicated, for availability and redundancy reasons. Let  $u_n$  be the unit column vector of  $n$  elements (all its entries are 1),  $s = (s_1, \dots, s_K)^t$  the video sizes and  $S = (S_1, \dots, S_P)^t$  the super-peers' storage capacity.

We define the Multi-Knapsack Double Set-Cover (MK-DSC) in matrix form as follows:

$$\begin{aligned} \min_E \sum_{j=1}^K \max \left\{ \frac{\lambda_j s_j}{\theta s_j + c}, \frac{\lambda_j s_j - \rho z_j}{\theta s_j + \eta \mu} \right\} \\ \text{s.t.} \\ E \times s \leq S \\ E^t \times u_P = z \\ z \geq 2u_K \\ E(p, j) \in \{0, 1\}, \forall p \in [P], j \in [K]. \end{aligned}$$

The objective is to minimize the mean waiting times among all video contents. The first constraint states that super-peer's storage capacity cannot be exceeded. The second constraint

relates the number of replicas for video content  $j$ , named  $z_j$ , with the matrix  $S$  (summing its columns). Finally, the third constraint imposes that each video content must be available in the network at least twice. Observe that the objective function depends on the variable  $z_j = \sum_p E(p, j)$  if and only if the maximum function is dominated by its second argument. This is true if and only if:

$$\frac{\lambda_j s_j - \rho z_j}{\theta s_j + \eta \mu} > \frac{\lambda_j s_j}{\theta s_j + c}. \quad (15)$$

Inequality (15) can be re-written to obtain:

$$2 \leq z_j < \frac{\lambda_j c - \eta \mu}{\rho c + \theta s_j}, \quad (16)$$

where we added the constraint  $z_j \geq 2$ .

The following remarks are corollary of Expression (16), and help to understand the limits in the design of a peer-assisted on-demand service:

- 1) Unless the download capacity  $c$  is high enough, the peers contribution to the system can be neglected without sacrificing performance. It is common in practice to find peers with  $c > 4\mu$ .
- 2) If the super-peers' capacity  $\rho$  is extremely high in relation with the peers needs (popularity  $\lambda_j$ ), the peers cooperation can be neglected again.
- 3) The decision variable  $z_j$  is also upper-bounded. In fact, the streaming rate is divided in the down-link once a new video content is included in the super-peers' storage. Inequality (16) represents a threshold for  $z_j$ .

We will solve the MK-DSC problem under different scenarios in order to show the strength and limitations of a P2P assisted on-demand video service.

### B. Greedy Randomized Resolution

The Multi-Knapsack Problem (MKP) is strongly NP-Hard [11]. Our problem has a similar flavor, but we must cover each item twice, and the profits are related with several parameters. We present a greedy randomized resolution for the MK-DSC. The metaheuristic can be studied in two stages. The first one constructs a seed of our GRASP-heuristic, and it is named *GreedySeed*. The second stage is a classical local search improvement, named *LocalSearch*.

In *GreedySeed* every content is greedily stored in the two *fattest* super-peers (i.e. the ones with the highest remaining storage capacity). Note that in this multi-knapsack flavored problem the costs are the content's sizes  $s_j$ , whereas the profits are the reduction of the population sizes  $\bar{x}_j$ .

Then, we introduce the will vector  $W = (w_1, \dots, w_K)$  such that

$$w_j = \frac{1}{s_j \bar{x}_{jP2P}^{SFM}}.$$

Note that the population size  $\bar{x}_{jP2P}^{SFM}$  depends on the number of super-peers  $z_j$  seeding video content  $j$ , which *a priori* is

unknown. For that reason, we compute first an approximation  $W' = (w'_1, \dots, w'_K)$  for the will vector  $W$ :

$$w'_j = w_j|_{z_j=0} = \min \left\{ \frac{\theta s_j + c}{\lambda_j s_j^2}, \frac{\theta s_j + \eta \mu}{\lambda_j s_j^2} \right\} \quad (17)$$

In practical network the peer's upload capacity is always lower than its download, so  $\eta \mu < c$ , and:

$$w'_j = \frac{\theta s_j + \eta \mu}{\lambda_j s_j^2} \quad (18)$$

Without loss of generality, we will assume that  $w'_1 > w'_2 > \dots > w'_K$  (in other words, videos are numbered in decreasing will when  $z_j = 0$ ).

*GreedySeed* is specified in Algorithm 1. The vector  $W'$  is computed in Line 1, using Equation (17). The video contents are sorted in decreasing will, in Line 2. In the iterative block (Lines 3 to 8) each video content is assigned in turns to the two fattest super-peers, named  $p_1$  and  $p_2$ . Video content  $j$  is then stored in both super-peers: the decision variables  $E(p_1, j)$  and  $E(p_2, j)$  are turned-on in Lines 5 and 6 respectively. Finally, the super-peer resources  $\{S_i\}_{i=1, \dots, P}$  are updated in Line 7. Algorithm *GreedySeed* returns a feasible solution, contained in the decision matrix  $E$ .

---

**Algorithm 1**  $E = \text{GreedySeed}(\lambda, \theta, \gamma, \eta, s, S, \mu, c, \rho)$

---

```

1:  $W' \leftarrow \text{FindWill}(\lambda, \theta, \gamma, \eta, s, \mu, c, \rho)$ 
2:  $\text{SortVideos}(W')$ 
3: for  $j = 1$  TO  $K$  do
4:    $(p_1, p_2) \leftarrow \text{TwoFattest}(S_1, \dots, S_P)$ 
5:    $E(p_1, j) \leftarrow 1$ 
6:    $E(p_2, j) \leftarrow 1$ 
7:    $\text{Update}(S_1, \dots, S_P)$ 
8: end for
9: return  $E$ 

```

---

In order to improve the solution  $E$  returned by *GreedySeed*, a local search improvement is introduced in a second stage. The idea is very simple: in each step we first try to add a new video replica. If no replica added then we try to delete an existing replica. If no replica added or removed then finally we try to swap two video replicas from randomly chosen super-peers. Each step takes effect only if the movement produces a better-and-feasible solution. The pseudo-code for this local search stage is shown in Algorithm 2.

---

**Algorithm 2**  $E^* = \text{LocalSearch}(E)$

---

```

1:  $(E, \text{improve}) \leftarrow \text{Add}(\text{Rand}(SP, \text{Video}))$ 
2: IF  $\text{improve}$  GO TO Line 1
3:  $(E, \text{improve}) \leftarrow \text{Delete}(\text{Rand}(SP, \text{Video}))$ 
4: IF  $\text{improve}$  GO TO Line 1
5:  $(E, \text{improve}) \leftarrow \text{Swap}(\text{Rand}(SP1, V1), \text{Rand}(SP2, V2))$ 
6: IF  $\text{improve}$  GO TO Line 1
7: return  $E$ 

```

---

**Remark V.1.**

- 1) *Functions Add, Delete and Swap work only if the new solution is both feasible and better in terms of the objective function.*
- 2) *The effects of Functions Add and Delete never cancel.*

## VI. RESULTS IN A REAL-LIFE SCENARIO

Currently, GoalBit supports high-quality Live and on-Demand video streaming to end users. We wish to improve the performance of the VoD distribution by adding or removing video replicas in the system. In order to predict the behavior of our new storage-scheduling technique, we picked up real-life traces taken from YouTube. A PHP-script was designed to collect some useful information as follows:

- (1) We take a video URL to start.
- (2) From this URL we get useful video data (size, time online, number of views, and others).
- (3) We save this data in a database.
- (4) We collect all the related videos URLs, and
- (5) Go back to Step (1) with a new video URL.

This process was executed during 3 days, allowing us to have useful information of more than 50.000 videos. With this information we estimated *videos' popularities*  $\lambda_j$  based on the number of views and the time online. We stress the system introducing a factor  $\beta$  to the vector  $(\lambda_1, \dots, \lambda_K)$ . In this way, we can contrast the performance of a CDN vs P2P deployment in flash-crowded, low-populated and intermediate scenarios. We use an abortion rate of  $\theta = 0,1$  peer per second, file sharing efficiency of  $\eta = 0,5$ , download rate of  $c = 1$  MegaBytes per second,  $d = c/4$ , and a system with  $P = 4$  super-peers (or servers) with a capacity of  $\rho = 10$  MegaBytes per second, storing  $K = 59000$  video contents. Figure 2 shows the estimated download time for the CDN and P2P models (blue and red lines respectively) versus  $i$ , where the stress factor  $\beta$  takes values  $10^i$ ,  $i = 1, \dots, 15$ . First, it is worth noticing that the expected time for a P2P sequential system is never worse than the one of a traditional CDN system, as can be predicted by Proposition IV.4. Second, the performance of both systems is quite similar for low-populated scenarios. Also, the time savings for peers are remarkable in high-populated scenarios. Finally, from last figure we can conclude that P2P system can work similarly with less resources, while CDN has a very important variability in its performance when increasing the number of servers. The results suggest that peers can download the desired video content more than five times faster than users in a traditional CDN, in massive scenarios.

A second experiment was conducted to figure out how the system's performance can be affected in terms of scalability. For a fixed popularity factor we want to find the mean waiting times for different number of super-peers (servers). Figure 3 illustrates the average waiting time for both P2P and CDN systems (red and blue lines respectively) versus  $P$ , where  $P$  is the number of super-peers (servers) in the system. We

fixed the popularity factor  $\beta = 10^3$ , but a similar behavior can be appreciated for other popularities. This suggests that the average waiting time in the P2P system is consistently low, whereas the CDN performance is effectively improved distributing the load to more servers. All test were executed in a home-PC (Intel Core i7, 8 GB RAM), getting more than 300.000 modifications during the *LocalSearch* phase, with a running-time of 14 hours for each experiment.

Fig. 2. Download time for CDN and P2P when increasing popularity

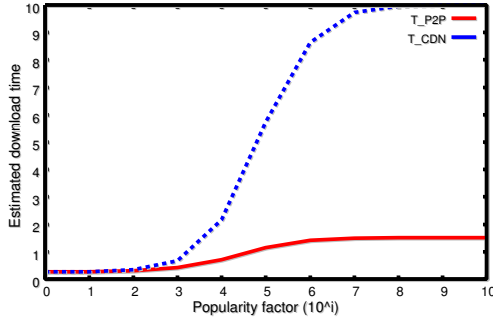
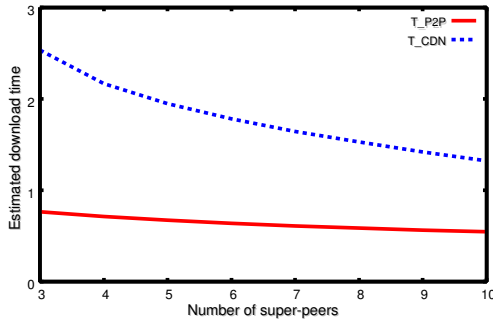


Fig. 3. Download time for CDN and P2P versus the number of node-servers



## VII. CONCLUSIONS AND FUTURE WORK

In this work we presented a mathematical framework for the analysis of concurrent and sequential video on-demand assisted applications. Under this framework, the P2P system reaches a stationary state in real-life scenarios, and outperforms traditional CDN in both concurrent and sequential services. We proved asymptotic and global stability of the system for particular scenarios. We presented necessary and sufficient conditions for the Sequential Fluid Model to be asymptotically stable, and conjectured it is globally stable whenever it is asymptotically stable.

An experimental validation of the P2P and CDN systems and their performance is presented considering real-traces taken from YouTube. The results are encouraging, showing that a P2P platform can perform much more better than traditional CDN in highly populated scenarios. Both systems are globally stable for sequential services.

We are aware that there are several lines that deserve further research. The global stability analysis is partially covered. The inclusion of peers heterogeneity and free-riding would introduce realism to the models. Here we presented a static

storage-allocation resolution, but a dynamic allocation in a controlled system can be very useful as well. Our trends for future work include stability and capacity analysis in video on-demand assisted scenarios (for both sequential and concurrent services), and the implementation of concurrent services in the GoalBit platform.

## REFERENCES

- [1] B. Cohen, "Incentives build robustness in bittorrent," *www.bramcohen.com*, vol. 1, pp. 1–5, May 2003.
- [2] "Bittorrent protocol specification v1.0," <http://wiki.theory.org/BitTorrentSpecification>, 2010.
- [3] M. E. Bertinat, D. D. Vera, D. Padula, F. Robledo, P. Rodríguez-Bocca, P. Romero, and G. Rubino, "Goalbit: The first free and open source peer-to-peer streaming network," in *Proceedings of the 5th international IFIP/ACM Latin American conference on Networking (LANC'09)*. New York, USA: ACM, September 2009, pp. 83–93.
- [4] GoalBit - The First Free and Open Source Peer-to-Peer Streaming Network, <http://goalbit.sf.net/>, 2008.
- [5] G. de Veciana and X. Yang, "Fairness, incentives and performance in peer-to-peer networks," October 2003.
- [6] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *Proceedings of SIGCOMM'04*, ACM. New York, NY, USA: ACM, September 2004, pp. 367–378.
- [7] D. Qiu and W. Sang, "Global stability of peer-to-peer file sharing systems," *Comput. Commun.*, vol. 31, no. 2, pp. 212–219, Feb. 2008.
- [8] Y. Tian, D. Wu, and K.-W. Ng, "Analyzing multiple file downloading in bittorrent," in *Proceedings of ICPP'06*. IEEE, August 2006, pp. 297–306.
- [9] P. Rodríguez-Bocca and C. Rostagnol, "Optimal download time in a cloud-assisted peer-to-peer video on demand service," in *Proceedings of the International Network Optimization Conference (INOC'11)*. London, UK: Springer, Lecture Notes in Computer Science, 13-16 June 2011.
- [10] D. Luenberger, *Introduction to dynamic systems: theory, models, and applications*. Wiley, 1979.
- [11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.