



**HAL**  
open science

## A window-based, server-assisted P2P network for VoD services with QoE guarantees

Noé Torres-Cruz, Mario Rivero-Angeles, Gerardo Rubino, Ricardo Menchaca-Méndez, Rolando Menchaca-Méndez

► **To cite this version:**

Noé Torres-Cruz, Mario Rivero-Angeles, Gerardo Rubino, Ricardo Menchaca-Méndez, Rolando Menchaca-Méndez. A window-based, server-assisted P2P network for VoD services with QoE guarantees. *Mobile Information Systems*, 2017, 2017 (Article ID 2084684), pp.1-16. 10.1155/2017/2084684 . hal-01665853

**HAL Id: hal-01665853**

**<https://inria.hal.science/hal-01665853>**

Submitted on 17 Dec 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A window-based, server-assisted P2P network for VoD services with QoE guarantees

Noé Torres-Cruz, Mario E. Rivero-Angeles, Gerardo Rubino, Ricardo Menchaca-Mendez, Rolando Menchaca-Mendez  
ntorresc@ipn.mx, mriveroa@ipn.mx, gerardo.rubino@inria.fr, ric@cic.ipn.mx, rmen@cic.ipn.mx

**Abstract**—In this paper we describe a P2P network that is designed to support VoD services. This network is based on a video file sharing mechanism that classifies peers according to the window (segment of the video) that they are downloading. Such classification allows identifying peers that are able to share windows among them, so one of our major contributions is the definition of an efficient and easy-to-implement video file sharing mechanism. Considering that cooperation among peers can be insufficient to guarantee an appropriate performance of the system, we also propose that the P2P network must be assisted by upload bandwidth from servers; we complement this idea by defining a highly efficient resources distribution scheme that prioritizes peers that are in windows with resources scarcity (we called it *Prioritized Windows Distribution scheme*). Given the above described system and on the basis of a fluid model and a Markov chain, we developed a methodology that allows selecting the design parameters values of the network (e. g. windows sizes and minimum servers upload bandwidth), that satisfies a set of target values for QoE parameters (e. g. MOS as a function of initial delay).

**Index Terms** - Video-on-Demand, Peer-to-peer Network, Markovian model, Fluid model, Window-Based Peer Selection, Video QoE

## I. INTRODUCTION

Peer-to-Peer (P2P) networks have been widely analyzed in order to increase the capacity of systems by having nodes that cooperate to alleviate data traffic at the servers. Unlike conventional client/server systems which experience a stark performance degradation when the number of clients increases, P2P networks are able to scale much better because their capacity also increases with the number of users. Originally, P2P networks were designed to distribute files whose download times were not very restricted, since those files were suppose to be used only after their download completion; however in recent years a large body of research has been focused on analyzing the same kind of networks, but supposing that the distributed file is a video whose playback is initiated even if its download is still in progress. Services as Live IPTV and Video on Demand (VoD) have been analyzed in the context of P2P networks in [1], [2] and [3]–[11] respectively.

Particularly, VoD streaming has become widely popular since it considerably reduces the startup time or initial delay of the video playback process, since it is not necessary to wait for the complete download of the file, and it allows subscribers to select and playback a video, as well, as rewind, fast-forward, pause or stop such video at any time. These features had made

VoD services very attractive and nowadays they represent an important proportion of the current internet traffic. According to [12], VoD represented approximately 23% of the total traffic in mobile networks during 2015 in North America and it is expected that this trend will continue in future mobile networks [13].

Considering this, it is highly probable that P2P networks will be a key technology for the distribution of video content in the next generation of wireless communications, including the fifth generation of mobile systems (5G); indeed, several works have recently addressed this issue by proposing strategies to allocate resources in this context [14], [15] and in this paper we also analyze this kind of networks and propose a resource allocation scheme as a function of QoE parameters.

One protocol that has had a lot of impact in the development of P2P networks is BitTorrent. In this protocol, the main idea is to divide the files into many pieces called *chunks*. To download a file, peers exchange these chunks following some rules. The BitTorrent protocol differentiates two types of peers: *leeches*, which are peers that have a subset (possibly the empty one) of the chunks that compose the file; and *seeds*, which are peers that have downloaded the whole file and remain in the system to share their resources. Both leeches and seeds cooperate to upload the file to other leeches. Whenever a peer joins the system with the objective of downloading the file, it contacts a particular node called *tracker* which has the complete list of peers that have part or all the file's content. Then, the tracker returns a random list of potential peers that might share the file with the arriving peer. At this point, the downloading peer contacts the peers on the list and establishes which chunks it is willing to download from each peer it is connected with.

BitTorrent is not suited for VoD applications, since chunks are distributed over the network in no particular order, while VoD services require a certain order on the download process to be able to guarantee a low initial delay. However, BitTorrent can still be used for streaming VoD services by making relatively minor modifications. The window-based peer selection scheme described in [16], [17] is an example of such modifications.

In [16], [17] the following procedure is proposed. The set of ordered chunks that composed the video file is divided into  $N$  segments of equal size that we called *windows*. These windows are denoted by  $w_0, w_1, \dots, w_{N-1}$ , and they follow

the chunks' order in the stream. The basic idea is that peers that are downloading the file at close points in the stream are in general grouped in the same window. As a consequence, all leeches have accurate knowledge about the specific peers in possession of potential chunks to share. Moreover, leeches that are downloading the file at window  $w_i$  can download chunks from peers in any group downloading the file at window  $w_j$  for  $j > i$ . Conversely, the peers downloading the file in window  $w_i$  can serve any other peer downloading the file at window  $w_j$  for  $j < i$ . By enabling this peer selection strategy, downloading peers only have to know the current window of the peers rather than the chunks that each peer possesses. This facilitates the task of identifying the peers to connect to. For this window-based strategy, it is assumed that: a) Leeches begin the file downloading process at window zero; b) Leeches at window  $i$  do not leave the current window until all chunks in it are downloaded and c) Peers download any chunk in a given window with no predefined order. The information about the individual downloading progress of each peer in the system is registered at the tracker. As it can be noticed, this window-based strategy does not require major changes to the original BitTorrent system.

In order to effectively implement this window-based scheme it is imperative to provide a methodology that allows selecting the appropriate number of windows in the system, which in turn defines the number of chunks per window. This value plays an important role on the average initial delay of the video playback, which is a major performance parameter.

Another important metrics that are commonly used to measure the performance of a video playback are the probability that a pause occurs and the average duration of that pause. In order to guarantee an acceptable performance of the system, we develop a mathematical analysis that allows the system manager to guarantee a certain level of user satisfaction, also known as Quality of Experience (QoE).

Through numerical results we evaluate the analytical expressions derived in this work and we conclude that in order to guarantee an acceptable level of the QoE the system requires the use of additional bandwidth that has to be provided by the network manager. Indeed, relying solely on the bandwidth of peers can not guarantee an acceptable performance under some conditions that depend on the random nature of arrivals and departures of peers and the level of peers cooperation. Specifically, we calculate the exact amount of additional bandwidth required to achieve conditions that guarantee that all leeches in the system download the file at the maximum rate.

Evidently, the use of this additional bandwidth increases the implementation cost of the system. As such, we propose two novel schemes to reduce as much as possible the required extra bandwidth while maintaining the required QoE levels:

- First, we propose a novel chunk distribution scheme, named Prioritized Window Distribution (PWD), where the additional servers provide more bandwidth to leeches

in higher windows rather than a uniform distribution as it was originally proposed. The rationale behind this scheme is that leeches in lower windows are served by seeds and other leeches in higher windows, while leeches in higher windows are only being served by seeds and a small amount of leeches. As such, a uniform distribution assigns a small amount of resources to leeches in higher windows.

- Second, we propose the use of a different size for the initial window in order to reduce the initial playback delay. As such, this mechanism allows to guarantee a QoE level target and by adjusting the rest of the parameters the probability and duration of a playback interruption are also reduced.

Finally, we provide strict guidelines in order to establish the system parameters in such a way as to provide QoE guarantees in different system conditions.

The rest of the paper is organized as follows: Section II discusses some of the previous work in the area and makes a detailed comparison with our work. Then, section III presents in detail the window based system including the main assumptions and considerations. Section IV explains the assisted server bandwidth proposal in order to attain abundance conditions in the system including the prioritized window distribution scheme to reduce the amount of extra server capacity. Following this, in Section V we derive the probability distributions of initial delay and interruption duration. Then, the implications of the window-based scheme on the QoE level is discussed in Section VI. Building on this, we provide strict guidelines to attain such QoE levels considering different system conditions. We end this paper discussing relevant numerical results and conclusions.

## II. RELATED WORK

One of the earliest works that pointed out the advantages of complementing traditional client-server networks with P2P systems in the context of video services was [18]. There, the authors demonstrate how much upload bandwidth from servers can be reduced by implementing a hybrid network. In recent years, a great researching effort from different perspectives has been made on analyzing this kind of systems in order to make them more efficient.

Some works have been focused on defining efficient P2P networks topology, e.g. in [3] a hybrid tree-mesh topology is proposed. Other researchers have identified that a way to increase a P2P network capacity is by implementing strategies that efficiently distribute content (videos) among the population of peers, examples of this kind of works are [4]–[6]. Schemes that incentivize cooperation among peers to increase the upload network capacity has been proposed on works such as [6]–[8] and [20]. Finally, the main focus of some other researchings has been on defining schemes that determine which peers are the most appropriate to serve a downloader (given the network topology, the video distribution and a level

of cooperation among peers) in order to improve a QoS or QoE parameter; some examples of this kind of research are [7], [9], [10], [14], [15], [20]. The scheme described in this paper belongs to the last classification, because the proposed window-based scheme defined which peers are capable to provide service to a given downloader; however, our contribution is not limited to that, since in addition we propose an efficient scheme to distribute the server resources. To our knowledge, this kind of proposals has not been published so far.

On the other hand, different analytical tools have been used to model P2P networks, including fluid models. In [19] a fluid model is proposed to analyze P2P networks and it is used to calculate performance parameters such as the number of peers in steady state in the system, as well as the average time required to download a file; however, the proposed scenario in that work do not consider the specific features of VoD services. In [16], [17] the fluid model was applied to a VoD, P2P network, where the shared video is split into windows in order to simplify the chunks interchange among peers. In this paper we also analyzed a windows-based P2P network, but we additionally consider that it is assisted by servers bandwidth and that the size of the initial window is different from the remaining ones. In recent years, some other papers have also reported analyses of VoD services over P2P networks that are based on fluid model, including [5], [11], [14], [20].

Among the works that were mentioned in the two previous paragraphs, [5], [7], [11], [20] are the most related to ours. In [7], a window-based P2P network is also described, the authors consider the existence of three buffers and a different strategy must be used to upload chunks to the network from each of them. The reason to propose such a scheme is the supposition that the peer storage capacity is limited; however, recent advances in hardware technology make low-priced devices increasingly equipped with abundant memory [20], and consequently in our system we propose the existence of only one buffer, which allows us to propose a sharing mechanism that is significantly simpler than the one described in [7].

In [20], it is presented a system which achieves scheduling video sharing between peers by adopting a dynamic buffering-progress-based, i. e. a downloader receives chunks only from peers with a similar playback progress; though this scheme has some similarities with the window-based scheme that we propose (the chunk sharing mechanism is based on the download progress of peers), and in both works a fluid model is used, the analysis perspective is quite different, e. g. where they propose that leeches stay in the system until the downloading is finished, we use a more realistic model in which leeches can leave the system at certain rate (denote by  $\theta$ ). Additionally, in that work, as well as in ours, one of the main targets is to reduce the required server bandwidth; however, we are interested in reducing it by proposing an efficient distribution among peers (PWD scheme), rather than incentivizing cooperation, as they propose.

In [11], the authors propose a modeling framework to compute the required server bandwidth, which has several similarities to our modeling: Poisson arrival process, analysis restricted to only one video, homogeneous download rate. Though the model that they propose considers a wider range of scenarios (non-stationary traffic, heterogeneous upload rate) than ours; our contribution isnt limited to compute the required server bandwidth, we also propose the aforementioned PWD scheme and evaluate its effect on the amount of required server bandwidth; hence we consider that these works can be complementary.

Additionally, in [11] the server bandwidth is computed considering that no interruptions occur during the playback process; however, a more significant reduction of those resources could be achieved by allowing the occurrence of initial delays and interruptions, provided that their durations and occurrence probabilities dont degrade significantly the users experience (measure trough QoE parameters); the integration of such a consideration, as well as its analysis, is one of the major contributions of this paper, since none of the above described works has addressed it and, to our knowledge, little research has been done about it. In order to consider the effect of QoE parameters, we use the experimental results reported in [21], where relations between QoS parameters (initial delays and interruptions duration) and QoE parameters (Medium Opinion Score, widely known as MOS) are established.

### III. THE MODEL FOR THE WINDOW-BASED SYSTEM

In this section we present the mathematical model for the *conventional* window-based strategy proposed for chunk selection download focused on stored streaming video services. We refer to this system as *conventional* since it relays only on the bandwidth shared by the peers in the network as proposed to [16], [19]. As it is shown below, the QoE guarantees in this system can be achieved only in very particular conditions, outside the capabilities of the network manager, since he can only relay on the cooperation among users. Also, this section presents the main assumptions and parameters of the system.

For the sake of facilitating the reading of this section as well as the following ones, we summarize the most relevant variables that are used throughout this paper, in Table I at the end of Section VI.

The number of leeches downloading in window  $w_i$  at time  $t$  is denoted by  $x_i(t)$ . The total number of leeches in the system at time  $t$  is denoted by  $x(t)$ , that is:

$$x(t) = \sum_{i=0}^{N-1} x_i(t). \quad (1)$$

There is a single file in the system, assumed for simplicity to be of size 1 as in [19]. The number of seeds in the system at time  $t$  is denoted by  $y(t)$ . Seeds share all chunks with leeches, independently of their current window. Additionally, according to the window-based strategy a leech can send all of its chunks to any leech on previous windows. We also

assume that at any given time there is at least one seed in the system, in order to prevent the starving of the system. Also, we assume that new leeches arrive at window 0 according to a Poisson process with rate  $\lambda$ . The transition rate from any peer at window  $i$  to the immediate superior one is denoted by  $\tau_i$  and that a leech leaves the system before the completion of the download or the playback process with rate  $\theta$ , while a seed leaves the system with rate  $\gamma$ . We assume that all peers have the same physical characteristics. Specifically: they all have the same uploading bandwidth  $\mu$  (in files/sec, recall that the file size is 1). We denote by  $1/c$  the (mean) time needed to download the whole file without interruptions working at full capacity, so that  $c$  is the maximal download rate for any peer, in files/sec, where  $c \geq \mu$ . The mean time needed to download a window is thus  $(1/c)/N$ ; we denote by  $c_w = Nc$  the corresponding window download rate. In the same way,  $\mu_w = N\mu$ .

All peers have complete knowledge of the system, i.e., all peers know in which group any other leech is. In a BitTorrent system, this can be done at the tracker. As such, if the number of leeches and seeds is sufficiently high, all leeches download the file at the maximum window download bandwidth  $c_w$ . This condition is referred to as *abundance*. However, when there are not enough peers in the system, the leeches download at a lower bandwidth. This condition is referred to as *penury*.

Finally, it is important to note that, in order to simplify the analysis, it is considered that users always play the video file in order, i.e., once some user starts the video playback, the latter is not fast-forwarded. This is due to the fact that all users consider that leeches in window  $w_i$  have all previous chunks (from window  $w_0$  to window  $w_{i-1}$ ). If some user fast-forwards the video, it may not continue the download of the chunks corresponding to the parts of the video that was not played. Recall that since a managed network is considered, this particular simplification can be easily implemented in a commercial system. However, in a future work, the case where users can forward and rewind the video playback will be considered.

From the previous description, the evolution in time of the number of leeches in each window,  $x_i$ , and seeds,  $y$ , for the system satisfies:

$$x'_0(t) = \lambda - \theta x_0(t) - \tau_0, \quad (2)$$

$$x'_i(t) = \tau_{i-1} - \theta x_i(t) - \tau_i, \quad 1 \leq i \leq N-1, \quad (3)$$

$$y'(t) = \tau_{N-1} - \gamma y(t), \quad (4)$$

where

$$\tau_i = \min \left\{ c_w x_i, \mu_w x_i \left( \sum_{k=i+1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) \right\}. \quad (5)$$

for  $i = 0, 1, \dots, N-1$ . In the case of the last window,  $i = N-1$ , we have

$$\tau_{N-1} = \min \left\{ c_w x_{N-1}, \mu_w x_{N-1} \frac{y}{x} \right\} \quad (6)$$

These last equations, which are related to the transition of peers from window  $i$  to window  $i+1$  can be explained as follows: Note that in case of abundance, leeches in window  $w_i$  download the file at the maximum bandwidth  $c_w$ . However, when there are not enough peers in the system, the leeches download at a lower bandwidth which is described as follows. First of all, note that all peers upload to leeches in window  $w_i$  with bandwidth  $\mu_w$ . Secondly, seeds upload the file to all peers in the system. As such, all the upload bandwidth is distributed uniformly among all leeches. Therefore, the proportion of the upload bandwidth for leeches at window  $w_i$  is  $x_i/x$ . Finally, only the leeches in a posterior window  $k$  (from  $i+1$  to  $N-1$ ) can send chunks to leeches in window  $w_i$ . This upload bandwidth is distributed uniformly for all leeches in windows 0 to  $k-1$ . As such, the proportion of the upload bandwidth for leeches in window  $w_i$  is  $x_i / \sum_{j=0}^{k-1} x_j$ . Leeches in the last window only receive chunks from seeds.

Let us compute the equilibrium point of this dynamical system, denoted by  $(\bar{x}_0, \bar{x}_1, \dots, \bar{y})$ . The equilibrium is obtained by solving the system  $\{x'_0(t) = 0, x'_1(t) = 0, \dots, y'(t) = 0\}$  and by replacing  $x_i(t)$  by  $\bar{x}_i$  and  $y(t)$  by  $\bar{y}$ . The system is then analyzed and the abundance conditions are identified. Assuming thus abundance, that is, assuming that  $\tau_{i,i+1} = c_w x_i$ , we obtain, after some algebra:

$$\bar{x}_0 = \frac{\lambda}{\theta + c_w}, \quad \bar{x}_i = \frac{\lambda c_w^i}{(\theta + c_w)^{i+1}}, \quad \bar{y} = \frac{\lambda}{\gamma} \left( \frac{c_w}{\theta + c_w} \right)^N, \quad (7)$$

For  $1 \leq i \leq N-1$ . And the total number of leeches in equilibrium, computed by:

$$\bar{x} = \sum_{i=0}^{N-1} \bar{x}_i = \frac{\lambda}{\theta} \left[ 1 - \left( \frac{c_w}{\theta + c_w} \right)^N \right] \quad (8)$$

All numerical explorations showed convergence towards equilibrium (see below, in this section, the comparison with a Markov model as well as the numerical results that are shown in section VIII).

In the following, the conditions to achieve abundance in the system are identified. This is an important feature for the practical implementation of the window-based strategy. In particular, the seeds departure rate is an important variable considering that seeds already have all the chunks of the video file. Observing (5), it can be seen that in order to have abundance at window  $w_i$ , the following condition must be met (we omit time  $t$ ):

$$c_w x_i \leq \mu_w \left( \sum_{k=i+1}^{N-1} x_k \frac{x_i}{\sum_{j=0}^{k-1} x_j} + y \frac{x_i}{x} \right). \quad (9)$$

From this condition, we can find a value  $\gamma_i$  that represents the maximum seeds departure rate that guarantees abundance in window  $i$  (i. e.  $\gamma \leq \gamma_i$ , for all  $i$ ). Consider  $t = \infty$  and

substitute (7) in (9); after some algebra, we get:

$$\gamma_i = \frac{\frac{(\theta + c_w)c_w^N}{(\theta + c_w)^N - c_w^N}}{\frac{c_w(\theta + c_w)}{\theta\mu_w} - \sum_{k=i+1}^{N-1} \frac{c_w^k}{(\theta + c_w)^k - c_w^k}} \quad (10)$$

Rate  $\gamma$  is clearly a *manager* variable that could be controlled according to (10). In Fig. 1, the abundance condition for different values of the departure rate for a leech ( $\theta$ ) is presented. It can be seen that as departure rate for a leech increases, it is necessary to increment the time that seeds remain in the system, i.e., the departure rate should be also reduced, since there are fewer leeches sharing the file. Hence, by using (10), and by measuring the arrival and departure rates for a leech, the network manager has a tool to offer an acceptable performance of the system by allowing all leeches download at the maximum capacity. In other words, our analysis allows finding an appropriated value for this control variable. However, achieving the adequate value of  $\gamma$ , i.e., encouraging peers to remain sufficient time in the system after the complete file download has occurred, is not an easy task; since when the peer has already played or downloaded the file, it has no need to remain longer just to share the file. One possibility to cope with this problem is to introduce penalties or rewards to peers in order to encourage a cooperative behavior (as in [6]–[8], [16], [17], [20]). However, these mechanisms do not guarantee to achieve an acceptable QoE in the network since they still rely on the behavior of the users. In a managed system, the network manager can decide for users equipment to remain connected sharing chunks to other users. Nonetheless, users are able to shut down or disconnect their equipment. This option is outside the capabilities of the network manager to provide QoE guarantees

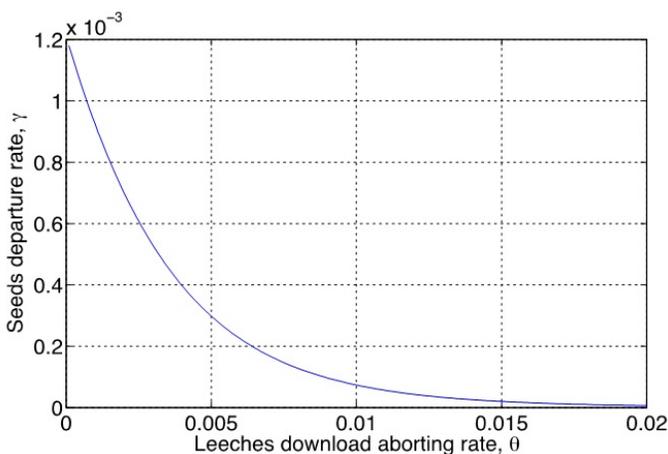


Fig. 1. Relationship between minimal departure rate for a seed and departure rate for a leeches in order to keep the system in abundance conditions;  $\lambda=10$ ,  $\mu=0.00125$ ,  $c=0.002$  and  $N=11$ .

From (10) it can also be observed than abundance condition

could be achieved by controlling the number of windows,  $N$ . Fig. 2 shows that by reducing  $N$ , the required value of  $\gamma_{N-1}$  to guarantee abundance is slightly relaxed (increased); however, reducing  $N$ , deteriorate the QoE parameters, as it is widely discussed in sections V to VII of this paper.

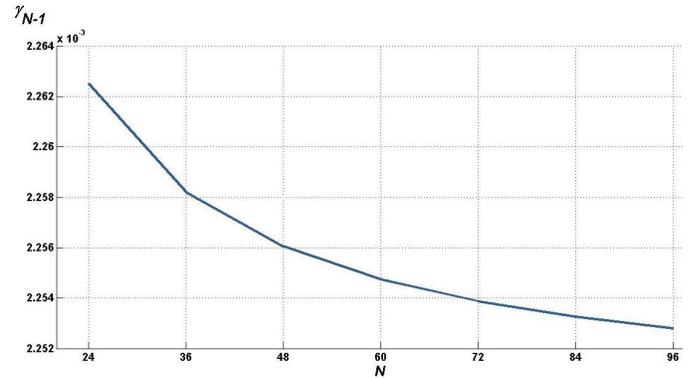


Fig. 2. Relationship between minimal departure rate for a seed and number of windows in order to keep the system in abundance conditions;  $\lambda=0.04$ ,  $c=0.00407$   $\mu=0.00255$  and  $\theta=0.001$ .

The rest of the variables that are involved in (10) are much harder to control, since the download and upload bandwidths ( $c_w$  and  $\mu_w$ , respectively) are usually fixed by the hardware used and the arrival rate ( $\lambda$ ) depends on the file popularity, i.e., how many users are willing to play the video.

Considering the previous paragraphs, we propose three mechanisms to improve the system's performance by guaranteeing the satisfaction of QoE parameters. First, we propose the use of additional server bandwidth that the network manager can explicitly designate to satisfy QoE parameters. Second, we propose an efficient distribution scheme of this extra servers resources, called *Prioritized Windows Distribution*, which is based on the window of the leech that is downloading the file. Third, we propose to use a different size for the first window in order to reduce the initial playback delay. These mechanisms can be fully controlled by the network manager and are explained in detail in the following sections.

We build now a discrete model of the same system. Consider vector  $W(t) = (L_0(t), L_1(t), \dots, L_{N-1}(t), S(t)) \geq (0, 0, \dots, 0, 1)$ , where  $L_i(t)$  and  $S(t)$  are respectively the number of leeches at window  $w_i$ ,  $i = 0, \dots, N-1$ , and of seeds, at time  $t$ . With all the previous "exponential assumptions",  $W(t)$  is a continuous time homogeneous Markov chain, with initial state  $W(0) = (0, 0, \dots, 0, 1)$ . Starting from state  $(l_0, l_1, \dots, l_{N-1}, m)$ , for any  $i \in \{0, 1, \dots, N-1\}$ ,  $l_i \geq 0$  and  $m \geq 1$ , the transition rates are described as follows:

- $\lambda$ , to state  $(l_0 + 1, l_1, \dots, l_{N-1}, m)$ ,
- $l_i\theta$ , to state  $(l_0, l_1, \dots, l_i - 1, \dots, l_{N-1}, m)$  ( $l_i \geq 1$ ),
- $\tau_{i-1,i}^*$ , to state  $(l_0, l_1, \dots, l_{i-1} - 1, l_i + 1, \dots, l_{N-1}, m)$  ( $1 \leq i \leq N-2$  and  $l_{i-1} \geq 1$ ),
- $\tau_{N-1,N}^*$ , to state  $(l_0, l_1, \dots, l_{N-1} - 1, m + 1)$  ( $l_{N-1} \geq 1$ ),
- $(m-1)\gamma$ , to state  $(l_0, l_1, \dots, l_{N-1}, m-1)$  ( $m \geq 2$ ),

where for  $0 \leq i \leq N - 2$ ,

$$\tau_{i,i+1}^* = \min \left\{ c_w l_i, \mu_w \left( l_i \sum_{k=i+1}^{N-1} \frac{l_k}{\sum_{j=0}^{k-1} l_j} + m \frac{l_i}{l} \right) \right\}$$

and

$$\tau_{N-1,N}^* = \min \left\{ c_w l_{N-1}, \mu_w \left( m \frac{l_{N-1}}{l} \right) \right\},$$

with  $l = \sum_{k=0}^{N-1} l_k$ .

#### IV. SERVER ASSISTED SYSTEM AND REDUCED INITIAL WINDOW

According to [21] the QoE that a user experiences during an on-line video playback is both related to initial delay and to probability that a pause occurs during the video playback, as well as the duration of such pauses. In this regard, it is important to note that there is a trade-off between the performance of these parameters that must be reached, since reducing the initial delay increases the probability that the video playback pauses and vice versa. The rationale behind this is as follows: when the device remains longer times downloading the file at the beginning of the video playback, a bigger portion of the file is available for its future playback, reducing the probability of a pause. From these parameters, it has been identified that the pausing probability and duration of pauses is much more harmful to the experience perception than the initial delay. Indeed, users prefer to wait longer times at the beginning of the file playback if it means that the video is not paused at all. However, this initial delay cannot be arbitrarily long since it would negatively impact the QoE of users. In the window-based strategy, the initial delay is directly related to the size of the windows. Note that it is not feasible to reduce the size of all windows in the system since the bandwidth required to attain abundance conditions are much harder with a large number of windows, as explained later in this section. However, the size of the first window can be established in order to guarantee an acceptable initial delay, while the rest of the windows can be set at the adequate value.

Summarizing, in order to have control over the aforementioned trade-off and provide an acceptable level of QoE, we propose two different mechanisms to improve the system performance:

- Unlike the basic-model described in the previous section, the size of the first window could be different from the size of the remaining windows. Specifically, it can be reduced in order to provide an acceptable initial delay. In order to prevent an increased video playback interruption probability we also implement the following mechanism.
- We propose that the P2P network to be assisted by a fixed download bandwidth provided by servers specifically used to provide QoE guarantees.
- A non-uniformly chunk distribution scheme is proposed in order to avoid the resource starving problem for

the leeches in upper windows. This scheme effectively reduces the required server assisted bandwidth to reach abundance conditions.

Taking into account the previous considerations, we denote by  $\mu_s$  the bandwidth which is provided by servers, and as an initial analysis we assume that this extra bandwidth is uniformly distributed among all the leeches, i. e. in the same way that the bandwidth provided by peers (both leeches and seeds) is distributed (as previously mentioned, in subsection IV-B we propose a more efficient bandwidth distribution scheme); and consequently, the upload bandwidth that servers can provide to leeches in window  $i$  is  $x_i \mu_s / x$ .

On the other hand, we denote by  $d_0$  and  $d_1$  the average time to download window 0 and window  $i$  (for  $1 \leq i \leq N - 1$ ), respectively. According to the definition of  $c_w$  we can establish that the download rates for window 0 and window  $i$  are respectively  $c_0 = 1/d_0$  and  $c_1 = 1/d_1$ . In order to simplify the subsequent equations, we define the parameter  $\alpha = c_0/c_1$  and after some algebraic manipulation we found that the aforementioned rates can be expressed as:

$$c_0 = c(\alpha N + (1 - \alpha)) \quad (11)$$

$$c_1 = c \left( N + \frac{1 - \alpha}{\alpha} \right) \quad (12)$$

Following an analogous analysis to the previous paragraph, we found that the upload rates for windows 0 and  $i$  are respectively given by:

$$\mu_0 = \mu(\alpha N + (1 - \alpha)) \quad (13)$$

$$\mu_1 = \mu \left( N + \frac{1 - \alpha}{\alpha} \right) \quad (14)$$

The incorporation of parameter  $\mu_s$ ,  $c_0$ ,  $c_1$ ,  $\mu_0$  and  $\mu_1$  in our analysis do not modify the essence of the fluid model and the Markov chain which were described in section III. However, the definition of  $\tau_i$  must be modified since in this case the rates to download or upload a window are given by equations (11-14) and, additionally, the upload capacity of the system must be increased by the server bandwidth. As a result of these new considerations, we can say that:

$$\tau_0 = \min \left\{ c_0 x_0, x_0 \left[ \mu_0 \left( \sum_{k=1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) + \frac{\mu_s}{x} \right] \right\} \quad (15)$$

$$\tau_i = \min \left\{ c_1 x_i, x_i \left[ \mu_1 \left( \sum_{k=i+1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) + \frac{\mu_s}{x} \right] \right\} \quad (16)$$

for  $1 \leq i \leq N - 2$ , and

$$\tau_{N-1} = \min \left\{ c_w x_{N-1}, x_{N-1} \frac{\mu_1 y + \mu_s}{x} \right\} \quad (17)$$

Following an analogous analysis to the one described in section III, we found that the new equilibrium point of the

system in abundance is now given by:

$$\bar{x}_0 = \frac{\lambda}{\theta + c_0}, \quad \bar{x}_i = \frac{\lambda c_0}{(\theta + c_0)} \frac{c_1^{i-1}}{(\theta + c_1)^i} \quad (18)$$

for  $1 \leq i \leq N - 1$ .

$$\bar{y} = \frac{\lambda \alpha}{\gamma \beta} \left( \frac{c_1}{\theta + c_1} \right)^N \quad (19)$$

Where  $\beta = \frac{\theta + c_0}{\theta + c_1}$ . And the total number of leeches in equilibrium is given by:

$$\bar{x} = \frac{\lambda}{\theta} \left( 1 - \frac{\alpha}{\beta} \left( \frac{c_1}{\theta + c_1} \right)^N \right) \quad (20)$$

It is important to note that even if the parameters related to the video file uploading ( $\mu_0$ ,  $\mu_1$  and  $\mu_s$ ) are not explicit in the previous equations, their values are fundamental to guarantee the abundance condition of the system, as it will be shown in the next sub-section. Finally, it must be said that (15-20) are reduced to their corresponding counterparts given by (5-8), when  $\alpha = 1$  (i. e. the sizes of all windows are equal).

#### A. Minimum server bandwidth requirement to guarantee abundance conditions

According to (15-20), the abundance condition can be expressed by:

$$c_0 \leq \mu_0 \left( \sum_{k=1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) + \frac{\mu_s}{x} \quad (21)$$

for window 0 and by:

$$c_1 \leq \mu_1 \left( \sum_{k=i+1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) + \frac{\mu_s}{x} \quad (22)$$

for windows  $1 \leq i \leq N - 2$ , and by:

$$c_1 \leq \mu_1 \frac{y}{x} + \frac{\mu_s}{x} \quad (23)$$

for window  $N - 1$ .

From (21) to (23), the abundance conditions in terms of  $\gamma$  can be found as shown in section III). However, since  $\gamma$  is a parameter that highly depends on users' behavior, we identify that a complementary way to guarantee abundance in the system can be based on  $\mu_s$ . As previously mentioned, we propose different schemes in order to guarantee abundance with a minimum value of  $\mu_s$ , (non-uniform chunk distribution and different size of the initial window). In this regard, it is important to notice that, as we show in our numerical evaluations, small values of  $\gamma$  significantly help to satisfy abundance conditions. As such, the use of incentives to encourage peers to remain longer times in the system is still an important issue to provide QoE guarantees.

If we define  $\mu_{min}^i$  as the minimum bandwidth that is required from servers in order to preserve the abundance condition in window  $i$ , we can say that such conditions are

guaranteed in the whole system if  $\mu_s \geq \mu_{min}^i$ , for every  $i$ . From (21) to (23) we obtained that:

$$\mu_{min}^0 = \max \left\{ 0, x \left[ c_0 - \mu_0 \left( \sum_{k=1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) \right] \right\} \quad (24)$$

$$\mu_{min}^i = \max \left\{ 0, x \left[ c_1 - \mu_1 \left( \sum_{k=i+1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) \right] \right\} \quad (25)$$

for  $1 \leq i \leq N - 2$ , and

$$\mu_{min}^{N-1} = \max \{ 0, c_1 x - \mu_1 y \} \quad (26)$$

In order to understand the intuition behind the *max* operation in (24-26), it must be observed that the network can reach abundance without the need of servers and in that case the second term of this operation will be a negative number. Building from this, it can be demonstrated that:

$$\mu_{min}^{i-1} = \max \left\{ 0, \mu_{min}^i - \frac{\mu_1 x x_i}{\sum_{j=0}^{i-1} x_j} \right\} \quad (27)$$

for  $2 \leq i \leq N - 1$ .

Since  $x_i$  and  $\mu_1$  are always non-negative values, it is clear that  $\mu_{min}^i \geq \mu_{min}^{i-1}$  for  $2 \leq i \leq N - 1$ . This inequality implies that if the bandwidth provided by servers is enough to reach abundance in window  $N - 1$ , then abundance is guaranteed in the remaining lower windows, i. e. the bandwidth required from servers to guarantee abundance in the whole system is  $\mu_{min} = \mu_{min}^{N-1}$ .

After substituting (19) and (20) in (26) we have that:

$$\mu_{min} = \max \left\{ 0, \frac{\lambda c_1}{\theta} \left[ 1 - \frac{\alpha}{\beta} \left( 1 + \frac{\theta \mu_1}{\gamma c_1} \right) \left( \frac{c_1}{\theta + c_1} \right)^N \right] \right\} \quad (28)$$

It is important to remark that when  $\alpha = 1$ ;  $\beta = 1$  and  $x_i$ ,  $x$ ,  $y$  and  $\mu_{min}$  we obtain the model described in section III.

From this model, we have identified an important issue that directly impacts the performance of the system. Specifically, we have noted that the uniform distribution of resources, produces a resource starvation for leeches in upper windows. Indeed, leeches at lower windows are served by leeches in upper windows, seeds, and the extra bandwidth that is provided by servers. However, leeches in upper windows, are served by a much lower amount of leeches. Additionally, there is a higher amount of leeches in lower windows than in upper windows, which, according to (27), produces that most resources are being consumed by leeches in lower windows. As such, the amount of resources assigned to peers in the upper windows is drastically reduced. The previous explanation is supported by Fig. 3, where the amount of upload bandwidth that a leech in window  $i$  can get is shown; it can be noticed that even with large amounts of server bandwidth (e. g.  $\mu_s = 3$ ) the peers in the upper windows can access only a small amount of resources. In order to have

a more efficient distribution we propose a novel chunk sharing scheme detailed in the following section.

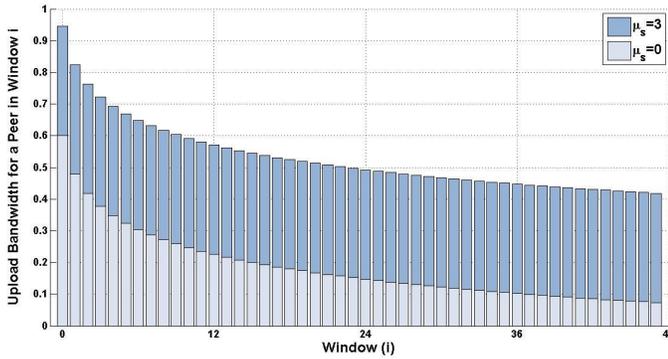


Fig. 3. Upload Bandwidth per Peer in Window  $i$ , for  $\lambda=0.04$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $\theta=0.001$ ,  $\gamma=0.006$ ,  $\alpha=1$  and  $N=48$ .

### B. Server bandwidth distribution scheme with prioritized windows.

In the scheme that was described in the previous subsection, the server bandwidth is uniformly distributed among all the leeches, which makes its implementation significantly simple. However, as mentioned above, a lot of extra bandwidth is required to provide abundance to leeches in the last window, since they have too few options to download their required chunks.

Consequently, we propose that the amount of server bandwidth that is assigned to the leeches in window  $i$  must be proportional not only to the numbers of leeches in that window, but also to an additional weight that must give priority to leeches in high windows. Specifically, we define that the server bandwidth assigned to leeches in window  $i$  must be proportional to  $x_i(i+1)^\varepsilon$ , since the factor  $(i+1)^\varepsilon$  will prioritize upper windows over the lower ones, for  $\varepsilon > 0$ . In the rest of the paper, this strategy is referred as *Prioritized Windows Distribution* (PWD) scheme, while the one described in Sub-section IV-A is referred as *Uniform Distribution* (UD) scheme.

According to the previous description, the transition rate from window  $i$  to window  $i+1$  now is given by:

$$\tau_0 = \min \left\{ c_0 x_0, x_0 \left[ \mu_0 \left( \sum_{k=1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) + \frac{\mu_s}{x_\varepsilon} \right] \right\} \quad (29)$$

$$\tau_i = \min \left\{ c_1 x_i, x_i \left[ \mu_1 \left( \sum_{k=i+1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) + \frac{(i+1)^\varepsilon \mu_s}{x_\varepsilon} \right] \right\} \quad (30)$$

for  $1 \leq i \leq N-2$ , and

$$\tau_{N-1} = \min \left\{ c_w x_{N-1}, x_{N-1} \left[ \frac{\mu_1 y}{x} + \frac{N^\varepsilon \mu_s}{x_\varepsilon} \right] \right\} \quad (31)$$

where  $x_\varepsilon = \sum_{j=0}^{N-1} (j+1)^\varepsilon x_j$  is a normalization used to guarantee that the sum of the server bandwidth assigned to all the windows has to be equal to  $\mu_s$ .

One important idea behind equations (29) to (31) is that when  $\varepsilon$  increases, the priority of window  $i$  over window  $i-1$  (for  $1 \leq i \leq N-1$ ) is more accentuated. If  $\varepsilon=0$ , these equations are reduced to (15)-(17). On the other hand, if  $\varepsilon < 0$ , the prioritized windows are the lower ones, which entails a system that assigns more resources to peers in low windows and less resources to peers in high windows, accentuating the resource starvation of these upper windows peers. Hence we are interested in analyzing the system only for  $\varepsilon > 0$ .

It is important to remark that these modifications in the server bandwidth distribution do not alter the essence of the previously described model, but only modifies the uplink capacity of the system, in the way that was already considered in equations (29)-(31).

The abundance conditions in terms of the server bandwidth now are given by:

$$\mu_0 = \max \left\{ 0, x_\varepsilon \left[ c_0 - \mu_0 \left( \sum_{k=1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) \right] \right\} \quad (32)$$

and

$$\mu_i = \max \left\{ 0, \frac{x_\varepsilon}{(i+1)^\varepsilon} \left[ c_1 - \mu_1 \left( \sum_{k=i+1}^{N-1} \frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x} \right) \right] \right\} \quad (33)$$

for  $1 \leq i \leq N-2$ , and

$$\mu_{N-1} = \max \left\{ 0, \frac{x_\varepsilon}{N^\varepsilon} \left[ c_1 - \mu_1 \frac{y}{x} \right] \right\} \quad (34)$$

It is important to emphasize that if  $\varepsilon$  is too large, the leeches in the higher windows would have access to an excessive amount of server bandwidth, while some other windows would become too prone to penury and consequently a lot of bandwidth server must be installed to guarantee abundance in those windows. Hence,  $\varepsilon$  must be selected in such a way that, given the parameters of the system, the abundance condition is guaranteed, while the amount of server bandwidth is maintained at the lowest possible value. In order to clarify the problem described above, in Fig. 4 we show  $\mu_{min}^i$  for different values of  $\varepsilon$ . It can be seen that with  $\varepsilon = 3$  the minimum assisted server bandwidth to guarantee abundance is  $\mu_{min} = 1.5$  and it is no longer for the last window,  $i = N$ . On the other hand, with  $\varepsilon = 1.5$ , the value of  $\mu_{min}$  is now less than 1.

Unlike the UD scheme analysis, in the PWD case it is not easy to determine which window requires the largest amount of server bandwidth and it is not straightforward to find a closed expression for the optimal value of  $\varepsilon$  ( $\varepsilon_{opt}$ ). However, we can find this optimal value by numerically evaluating :

$$i_{crit}(\varepsilon) = \operatorname{argmax}_i \{ \mu_i(\varepsilon) \} \quad (35)$$

$$\varepsilon_{opt} = \operatorname{argmin}_\varepsilon \{ \mu_{i_{crit}(\varepsilon)} \} \quad (36)$$

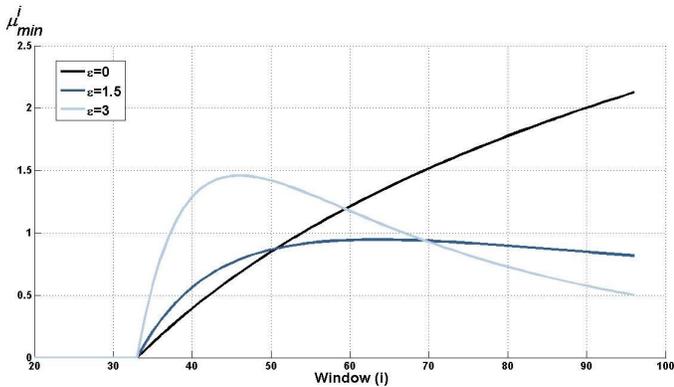


Fig. 4. Minimum server bandwidth required by window  $i$  for  $\lambda=0.04$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $\theta=0.001$ ,  $\gamma=0.006$  and  $N=96$ .

Where  $i_{crit}(\varepsilon)$  represents the index of the window that requires the largest value of  $\mu_i^{min}$  for a given  $\varepsilon$ . Strictly,  $\varepsilon_{opt}$  depends on all the parameters of the system. However, after evaluating (35) and (36), we found that it only significantly depends on  $\theta$  and  $\gamma$ , as it is shown in Fig. 5. Considering this, we propose an approximated calculation of  $\varepsilon_{opt}$ , which is given by:

$$\varepsilon_{opt} \approx 4916\theta^2 - 139\theta + 2279\theta\gamma - 49\gamma + 971\gamma^2 + 1.65 \quad (37)$$

The polynomial coefficients in (37) were obtained by apply-

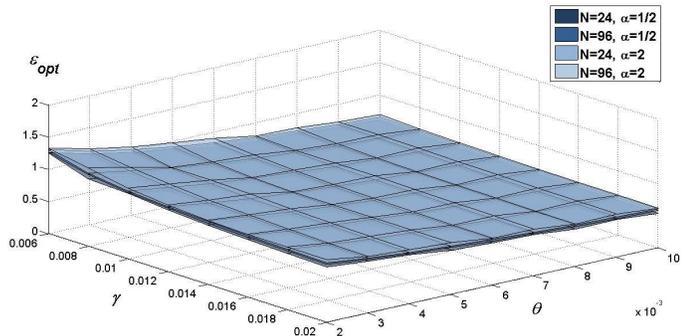


Fig. 5. Optimal values of  $\varepsilon$  for  $\lambda=0.04$ ,  $c=0.00407$  and  $\mu=0.00255$ .

ing a second order linear regression to the results shown in Fig. 5. In Fig. 6 we show a comparison between the exact evaluation and the corresponding approximation. Lastly, it must be noticed that even if  $\varepsilon$  is chosen exclusively in terms of  $\theta$  and  $\gamma$ ,  $\mu_{min}$  is still a function of the remaining parameters of the system (e. g.  $N$  and  $\alpha$ ). In Section VIII, we show that using the evaluation of  $\varepsilon_{opt}$  that is defined by (37), the PWD scheme significantly reduces  $\mu_{min}$  in comparison with the UD scheme.

## V. PROBABILITY DISTRIBUTIONS OF INITIAL DELAY AND INTERRUPTION DURATION

So far, we have described the operation of the proposed network, we have developed a mathematical model to evaluate

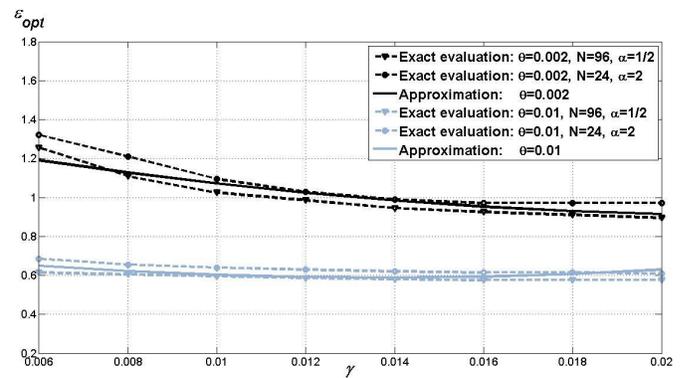


Fig. 6. Optimal values of  $\varepsilon$ : exact vs. approximated evaluations ( $\lambda=0.04$ ,  $c=0.00407$  and  $\mu=0.00255$ ).

the number of peers under abundance conditions and proposed two schemes to distribute additional bandwidth provided by servers. However, one of our major concerns in this paper is to establish a method to select the parameters of the system that satisfy some Quality of Experience (QoE) targets. To this end, we first analyze the behavior of the Quality of Service (QoS) parameters that are related to such QoE targets. Specifically, in this section we model the probability distributions of initial delay and interruption duration along the playback of a video which is being downloaded from the described system.

Since the mathematical analysis in the previous sections are valid only if the abundance condition exists, the analysis in this section is also limited to such circumstances, i. e. the probability distributions that we find are valid only if the system is in abundance. Additionally, in our analysis we are assuming that the playback of any window starts until that window has been completely downloaded, as it is shown in Fig. 7 (e. g. the video playback starts until window 0 has been completely downloaded). This assumption is valid because some buffering is indispensable to satisfy QoE targets that are related to interruptions. In Fig. 7 we also illustrate the meaning of some of the variables that are defined along this section.

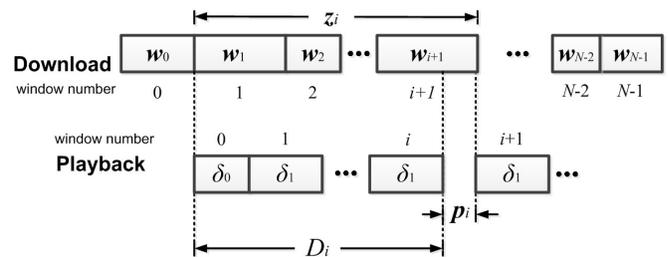


Fig. 7. Relationship between download and playback processes.

### A. Probability distribution of initial delay and mean downloading time

We are assuming that the distributions of the sojourn time of leeches and the time to download window 0 are both negative exponential, so they are given by  $f_u(x) = \theta e^{-\theta x} u(x)$  and  $f_{v_0}(x) = c_0 e^{-c_0 x} u(x)$ , respectively.

It is important to notice that according to our model,  $\mathbf{u}$  and  $\mathbf{v}_0$  are independent random variables and while the former models the users' sojourn time in the system, the latter only models the time to download window 0, with no regard of the user's sojourn. According to that, we define a new random variable,  $\mathbf{w}_0$ , which represents the time to download window 0, given that the sojourn time was large enough to achieve such download. Hence,  $\mathbf{w}_0$  is equal to  $\mathbf{v}_0$ , given that  $\mathbf{v}_0 \leq \mathbf{u}$ . After applying Bayes' theorem, it can be found that the distribution of  $\mathbf{w}_0$  is:

$$f_{w_0}(x) = (c_0 + \theta) e^{-(c_0 + \theta)x} u(x) \quad (38)$$

Under the assumption that the playback of window 0 will start until it has been successfully downloaded, we can say that (38) also represents the initial delay probability distribution. Furthermore, the mean initial delay is given by:

$$T_0 = \frac{1}{c_0 + \theta} \quad (39)$$

Note that  $T_0 < 1/c_0$  because  $T_0$  is an average download time that does not include the cases in which the downloading time is larger than the leech's sojourn time.

In order to find the distribution of the required time to download window  $i$  (for  $1 \leq i \leq N - 1$ ), it is necessary to identify the distribution of the remaining sojourn time of a leech that has downloaded the preceding windows. Let  $\mathbf{r}_0$  be the remaining sojourn time of a leech that has downloaded window 0. Hence,  $\mathbf{r}_0 = \mathbf{u} - \mathbf{v}_0$ , given that  $\mathbf{u} - \mathbf{v}_0 > 0$ . Considering the distributions of  $\mathbf{u}$  and  $\mathbf{v}_0$ , as well as Bayes' theorem it can be proved that:

$$f_{r_0}(x) = \theta e^{-\theta x} u(x) \quad (40)$$

As it can be observed, this distribution is identical to the distribution of  $\mathbf{u}$ . Although this result could seem a contradiction, it is explained by the fact that now we are only focused on those leeches whose sojourn time is large enough to successfully download window 0 and as a consequence of the memoryless property of the negative exponential distribution.

Given the previous result, it is clear that the distribution of the required time to download window 1, given that window 0 was downloaded, can be found by substituting  $c_0$  by  $c_1$  in (38). Furthermore, after applying the previous analysis to every window, we say that the distribution of the required time to successfully download window  $i$ , given that all the preceding windows were downloaded (denoted by  $\mathbf{w}_i$ , for  $1 \leq i \leq N - 1$ ), is:

$$f_{w_i}(x) = (c_1 + \theta) e^{-(c_1 + \theta)x} u(x) \quad (41)$$

and the mean time to download window  $i$ , given that the leech has not left the system at this point, is simply  $T_i = \frac{1}{c_1 + \theta}$ . Consequently the mean time to download the whole file, given that the leech did not leave the system is:

$$T = \frac{1}{c_0 + \theta} + \frac{N - 1}{c_1 + \theta} \quad (42)$$

It is interesting to note that if  $\alpha = 1$ , the required time to download the whole file has an Erlang distribution with rate parameter equal to  $c_w + \theta$  and shape parameter equal to  $N$ ; and (42) is reduced to  $T = \frac{1}{\theta/N + c}$ . In this expression it is clear that by increasing the number of windows  $T$  also increases but only to a maximum value of  $1/c$ .

### B. Probability distribution of interruption duration

On the basis of some of the previous results, in this subsection we find the probability distribution of the interruption duration that may occur after the playback of window  $i$ .

As it was previously established, we are considering that the playback of a window initiates until all the chunks of the window have been completely downloaded; Building from this, an interruption can occur after the playback of window  $i$ , if the time required to playback all the windows from 0 to  $i$  is smaller than the time required to download windows 1 to  $i + 1$ .

Due to the different initial window size scheme proposed to guarantee the QoE, we can represent by  $\delta_0$  the playback time of window 0 and by  $\delta_1$  the playback time of any other window (notice that these variables are proportional to  $d_0$  and  $d_1$ , respectively). Hence, the playback time of windows 0 to  $i$  can be expressed by  $D_i = \delta_0 + i\delta_1$ . Additionally, let  $\mathbf{z}_i$  be the time to download windows 1 to  $i + 1$ . Hence  $\mathbf{z}_i = \sum_{1}^{i+1} \mathbf{w}_i$  (see Fig. 7). Considering (41), we know that  $\mathbf{z}_i$  will follow an Erlang distribution with rate parameter equal to  $c_1 + \theta$  and shape parameter equal to  $i + 1$ .

Now, let  $\mathbf{p}_i$  be a random variable that represents the interruption duration after playing window  $i$ . From Fig. 7,  $\mathbf{p}_i$  can be expressed by:

$$\mathbf{p}_i = \begin{cases} 0; & \mathbf{z}_i \leq D_i \\ \mathbf{z}_i - D_i; & \mathbf{z}_i > D_i \end{cases} \quad (43)$$

And consequently, its probability density function must be given by  $f_{p_i}(x) = F_{z_i}(D_i)\delta(x) + f_{z_i}(x + D_i)u(x)$ , where  $F_{z_i}(x)$  is the cumulative distribution function of  $\mathbf{z}_i$  and  $\delta(x)$  is the Dirac's delta function. Then, the probability density function of the interruption duration after playing window  $i$  is given by:

$$f_{p_i}(x) = \left[ 1 - \sum_{j=0}^i \frac{e^{-(c_1 + \theta)D_i} ((c_1 + \theta)D_i)^j}{j!} \right] \delta(x) + \frac{(c_1 + \theta)^{i+1} (x + D_i)^i}{i!} e^{-(c_1 + \theta)(x + D_i)} u(x) \quad (44)$$

Notice that the definition of  $\mathbf{p}_i$  implies that no interruptions occurred at the end of the preceding windows ( $D_i$  is a

deterministic variable that only models the playback time). Additionally, we are considering that the user did not pause or move forward or backward the video. Despite these limitations, the distribution which is defined in (44) can be very useful to define QoE targets as a function of the parameters of the system, as we will see in the following section.

## VI. QOE AS A FUNCTION OF INITIAL DELAY AND INTERRUPTION DURATION IN YOUTUBE SERVICE

Several works have identified that conventional QoS parameters (e. g. bandwidth, jitter, delay), not necessarily are correlated to users experience. Hence significant effort has been made in order to link them to QoE parameters ( [21], [24]).

Particularly, in [21] some experiments were conducted in order to define MOS (one of the most used QoE parameters) as a function of initial delay and duration of interruptions that occurs along the playback of YouTube videos, one of the most popular service of VoD.

In regard of initial delay, [21] performed different measurements and expressed the MOS as a function of initial delay ( $w_0$ ). Specifically, it is modeled by  $M_{id}(w_0) = 5 - 0.862 \log w_0 + 6.718$ , where the highest MOS that can be reached is 4.287, with  $w_0 = 0$ . Since in our model the video playback starts until window 0 has been downloaded, we can evaluate the previous equation by substituting  $w_0$  by  $w_0$ . Moreover, if we define an initial delay target MOS, denoted by  $M_{id,t}$ , we can define a corresponding target initial delay as:

$$w_t = 10^{\frac{5-M_{id,t}}{0.862}} - 6.718 \quad (45)$$

Finally, we can evaluate the probability that  $M_{id,t}$  is not satisfied, since we know the distribution of  $w_0$ :

$$q_{id} = P \{M_{id}(w_0) < M_{id,t}\} = P \{w_0 > w_t\} = e^{-(c_0+\theta)w_t} \quad (46)$$

In [21] the authors also described the MOS as a function of the interruption duration ( $p_0$ ), with the following expression  $M_{int}(p_0) = 1.75e^{-0.334p_0} + 3.19$ . In this case, we can also define a target MOS, denoted by  $M_{int,t}$ . Hence a target interruption duration can be defined as:

$$p_t = -\frac{1}{0.334} \ln \left( \frac{M_{int,t} - 3.19}{1.75} \right) \quad (47)$$

At this point, it is important to remark that the aforementioned MOS model was designed considering only one interruption along the video playback. However, the model for interruptions that we developed in Section V considers interruptions at the end of every window; which means that only one of the random variables  $p_i$  can be used at one time to evaluate an analogous probability to (46). We select  $p_0$  because it has the advantage that does not depend on other interruptions (since it is the first one) and, most important, this is the most probable pause, as it can be demonstrated from (44) and under the assumption that  $\delta_0 \leq d_0$  and  $\delta_1 \leq d_1$ .

Considering the previous paragraphs, we can also evaluate the probability that  $M_{int,t}$  is not satisfied, since we know the distribution of  $p_0$ , i. e.:

$$q_{int} = P \{M_{int}(w_0) < M_{int,t}\} = P \{p_0 > p_t\} = e^{-(c_1+\theta)(\delta_0+p_t)} \quad (48)$$

It is relevant to make a comparison between (46) and (48) in terms of the size of window 0. It can be seen from (46) that when  $c_0$  increases (the size of the window decreases), the probability that  $M_{id,t}$  is not satisfied is reduced; while, under the same assumption, the probability of no satisfying  $M_{int,t}$  is increased (since  $\delta_0$  and  $c_1$  decrease). In other words,  $c_0$  and  $c_1$  must be selected in such a way that a trade-off between these QoE parameters exists. An alternative interpretation of this trade-off is that when the initial buffering is small, the user has a high probability of perceiving a satisfying small initial delay; but this, inherently, increases the probability of perceiving an unsatisfying interruption.

Having said that, we consider that in order to have a complete set of QoE parameters (and elements to properly select  $c_0$  and  $c_1$ ), it is needed to define target probabilities that  $M_{id,t}$  or  $M_{int,t}$  are not satisfied, which are denoted by  $q_{id,t}$  and  $q_{int,t}$ , respectively. According to (46) and (48), it can be said that we want that  $q_{id} = e^{-(c_0+\theta)w_t} \leq q_{id,t}$  and that  $q_{int} = e^{-(c_1+\theta)(\delta_0+p_t)} \leq q_{int,t}$  which lead us to:

$$d_0 \leq \frac{-w_t}{\ln(q_{id,t}) + \theta w_t} \quad (49)$$

$$d_1 \leq \frac{-(\delta_0 + p_t)}{\ln(q_{int,t}) + \theta(\delta_0 + p_t)} \quad (50)$$

As it will be shown in the next sub-section, (49) and (50) are used to select the number of chunks in every window in such a way that the QoE parameter set be satisfied.

Another important effect that may be observed while selecting  $d_0$  and  $d_1$  is that while  $M_{id}(w_0)$  is not very susceptible to increases in the initial delay, even small increases in the interruption duration may seriously degrade  $M_{int}(p_0)$  (since  $M_{id}(w_0)$  is logarithmic, while  $M_{int}(p_0)$  is exponential). According to [21], this is due to the fact that users are more tolerant of long initial delays than they are of long interruptions.

## VII. PARAMETER SELECTION FOR YOUTUBE SERVICES

In order to evaluate the performance of the proposed system, we are considering some currently implemented features of Youtube service by using the measurements reported by [22] and [23]. According to [22] the download strategies that are used to download a video in Youtube depend on the network and the device that are involved. Hence, it is relevant to specify that the measurements they are reporting were obtained with mobile devices downloading videos through a WiFi network.

Parameter	Description
$c$	Download rate for the complete file ( $c \geq \mu$ )
$c_0$	Download rate for the initial window
$c_1$	Download rate for window $i$ , $i > 0$
$M_{id,t}$	Target MOS as function of initial delay
$M_{int,t}$	Target MOS as function of interruption duration
$N$	Number of windows
$q_{id}$	Probability that $M_{id,t}$ is not satisfied
$q_{int}$	Probability that $M_{int,t}$ is not satisfied
$T$	Required time to download the video file
$T_0$	Required time to download the initial window
$x$	Number of leeches in steady state
$x_i$	Number of leeches in window $i$ in steady state
$y$	Number of seeds in steady state
$\alpha$	Non-initial window size to initial window size ratio
$\gamma$	Departure rate for a seed
$\varepsilon$	Priority control parameter
$\theta$	Departure rate for a leech
$\lambda$	Arrival rate of peers
$\mu$	Upload rate for the complete file
$\mu_0$	Upload rate for the initial window
$\mu_1$	Upload rate for window $i$ , $i > 0$
$\mu_{min}$	Minimum servers upload bandwidth that guarantees abundance in the whole system
$\mu_{min}^i$	Minimum servers upload bandwidth that guarantees abundance in window $i$
$\mu_s$	Servers upload bandwidth
$\tau_i$	Transition rate from window $i$ to window $i+1$
$\tau_{N-1}$	Transition rate from window $N-1$

TABLE I  
MOST RELEVANT VARIABLES SUMMARY

We now give a brief explanation on how some YouTube service parameters are related to our model. According to [22], the most common video format in YouTube service (MPEG-4 Visual) has an encoding rate between 200 and 275 kb/s, and the authors also identify that the download rate is allowed by this service is two times the encoding rate. If we denote by  $r_{cd}$  the ratio between the download rate and the encoding rate, we have  $r_{cd} = 2$  for the aforementioned case.

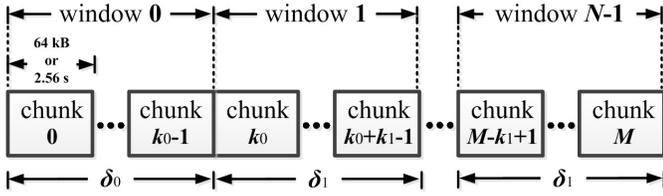


Fig. 8. Evaluation reference file.

In addition, in [23] it is reported that the average YouTube video duration is 490.5 seconds. Considering this, a codification rate of 200 kb/s, as well as the fact that in YouTube the chunk size is 64kBytes, we select as a reference to our evaluation a video file whose features are illustrated in Fig. 8, where  $M$  is the number of chunks in the file. As such, according to the previous data, our reference file is set at  $M = 192$ . Additionally we denote by  $k_0$  and  $k_1$  the number of chunks in window 0 and in any other window, respectively.

From the previous data it can also be established that the

normalized download rate for the reference file would be  $c = 0.00407$ . Because of the usual asymmetry in the users network access links, we consider  $\mu < c$ . Specifically we select a normalized upload rate  $\mu = 0.00255$  (which is equivalent to a data transmission rate of 250 kb/s, for our reference file). Given the reference file, the possible values for  $N$  are between 1 and 192; so we consider  $N = 24, 48, 60, 72, 84$  and 96 for most of our evaluations.

According to [23], the average playback time of a YouTube video before it is interrupted by the user is 172 seconds. Notice that in our model we identified this variable as the leeches' sojourn time. Hence  $\theta = 1/172 = 0.0058$ . However, in order to evaluate our system under a variety of scenarios, we set  $\theta$  for values from 0.002 to 0.01 (the superior limit corresponds to a sojourn time of 100 seconds).

To model the seeds sojourn time, we set  $\gamma$  in the range from 0.006 to 0.02; which accounts for seeds remaining in the system an average of 50 to 167 seconds.

Additionally, now that we have introduced the video file parameters, we can establish relations between them and some model variables. Specifically,  $d_0 = \frac{k_0}{cM}$ ,  $d_1 = \frac{k_1}{cM}$ ,  $\delta_0 = r_{cd}d_0$  and  $\delta_1 = r_{cd}d_1$ .

In the following we provide a method to select the design parameters of the system:

- First, it must be noticed that we are interested on satisfying a predetermined QoE parameter set, which means that we have to select  $d_0$  and  $d_1$  in such a way that inequalities (49) and (50) are simultaneously satisfied. Since (49) is only a function of  $\theta$  and QoE targets we can select the number of chunks in window 0 by:

$$k_{0,s} = \lfloor \frac{-w_t c M}{\ln(q_{id,t}) + \theta w_t} \rfloor \quad (51)$$

- Once that  $k_0$  is selected, an analogous calculation to (51) can be carry out to select  $k_1$  by using (50), i. e.:

$$k_{1,s} = \lfloor \frac{-cM(r_{cd}k_{0,s} + cMp_t)}{cM \ln(q_{int,t}) + \theta(r_{cd}k_{0,s} + cMp_t)} \rfloor \quad (52)$$

- From (51) and (52) we can define the selected values of  $N$  and  $\alpha$  by:

$$N_s = \lceil \frac{M - k_{0,s}}{k_{1,s}} \rceil + 1; \quad (53)$$

$$\alpha_s = \frac{k_{1,s}}{k_{0,s}} \quad (54)$$

Notice that  $k_{0,s}$  and  $k_{1,s}$  are the maximum values that satisfy the QoE parameters set, which means that in the case that lower values are used,  $N$  would acquire a larger value than  $N_s$ . However, as it is shown in Section VIII, the larger the values of  $N$ , the larger the values of  $\mu_s$  that are required to guarantee abundance.

Assuming that the users' behavior and download/upload parameters are known and using  $N_s$  and  $\alpha_s$ , it is possible to

numerically evaluate (36). This solution provides the optimal values of  $\varepsilon$  and  $i$ . By substituting these values in (32), (33) or (34), depending on the value of  $i$ , the minimum value of  $\mu_s$  that guarantees abundance can be calculated. Notice that the previous operations are needed only if the PWD scheme is used; if the UD scheme is used,  $\mu_{min}$  can be calculated by simply substituting  $N_s$  and  $\alpha_s$ , as well as the remaining parameters, in (28).

### VIII. NUMERICAL EVALUATION

In this section, we provide relevant numerical results that evaluate the performance of the system in terms of the required server capacity to guarantee QoE, the average initial delay, and the average download time for different system parameters. Additionally, we provide the required parameters to guarantee an acceptable level of QoE for different scenarios.

First, we show in Fig. 9 the minimum bandwidth that servers must provide in order to achieve abundance in all the windows, while making a comparison between the distribution schemes described in Sub-sections IV-A and IV-B. The values shown in this figure were found by converting the normalized rates (file length equal to one) into practical data rates, according to the values that were mentioned in Section VII. As it was previously explained, the PWD drastically reduces the extra capacity required to provide abundance conditions in all the system compared to the uniform distribution scheme. It is also important to mention that, as it was expected, when leeches are more cooperative ( $\theta$  takes small values), the value of  $\mu_{min}$  is smaller. Additionally, it is crucial to emphasize that increases in the number of windows significantly increases the required server bandwidth; however, as it is shown later, a very small number of windows has a negative impact on the QoE parameters. In addition to this, Fig. 10 shows  $\mu_{min}$  for two different values of  $\gamma$ . It can be seen that the existence of cooperative seeds (small values of  $\gamma$ ) significantly reduces the need of the server assisted bandwidth in the P2P network.

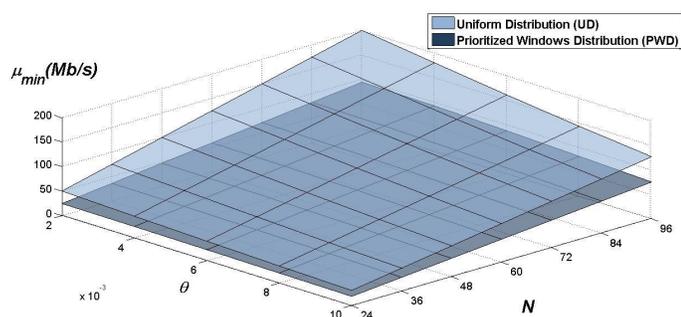


Fig. 9. Minimum server bandwidth to achieve abundance, considering  $\lambda=0.04$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $\gamma=0.006$  and  $\alpha=1$

In Fig. 11 - 14 we show the system performance in terms of the number of leeches ( $x$ ), the number of seeds ( $y$ ), the average initial delay ( $T_0$ ) and the average video download delay

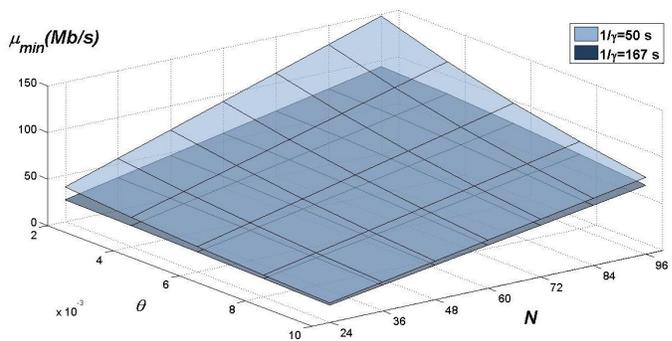


Fig. 10. Minimum server bandwidth to achieve abundance, considering the PWD scheme,  $\lambda=0.04$ ,  $c=0.00407$ ,  $\mu=0.00255$  and  $\alpha=1$

( $T$ ), respectively; which were obtained through the evaluation of (20), (19), (39) and (42) and by numerically solving the corresponding Markov chains. These evaluations were carried out using the PWD scheme, as well as a server capacity of 120 Mb/s, which is equivalent to a normalized download file rate of  $\mu_s=1.24$ . Also, abundance conditions are guaranteed. As it was expected, a small value of  $\theta$  entails a large number of leeches, as it is depicted in Figure 11. Since the seeds are peers that finish the file download, the number of them also increases when  $\theta$  decreases, as it is shown in Fig. 12. Indeed, these conditions corresponds to a cooperative system which, as previously mentioned, reduces the required assisted server bandwidth to guarantee abundance. Though these parameters are not directly related to the QoE levels, they offer an insight into the system performance. For example, it is clear that the larger the number seeds, the larger the capacity in the system.

In Fig. 13 we corroborate that in order to reduce the average initial delay,  $T_0$ , the size of the first window must be reduced. In these results, the initial window size is reduced by increasing  $N$ , although the rest of the windows are also reduced since we selected  $\alpha=1$ . Indeed, by increasing  $N$ , the number of chunks per window decreases, effectively reducing the initial delay. However, this has a negative effect in the overall system performance, as shown in Fig. 9 where a high value of  $N$  entails a higher amount of assisted bandwidth to maintain the system in abundance conditions. As such, in order to reduce the initial delay efficiently, the value of  $\alpha$  should be increased instead. Additionally, it must be noticed that since  $\theta$  is related to both system capacity and resources demand, it has no significant effect in  $T_0$ . These results exhibits the scalability properties of P2P networks.

Regarding the performance of average download delay,  $T$ , Fig. 14 presents some interesting results that show that under abundance conditions its value is almost a constant. The reason for this is that, in these conditions, the required time to download any window is the same (except for window 0, when  $\alpha \neq 1$ ), according to (42). It is important to note that results in Fig. 11 - 14 directly proved that our analytical solution agrees with the numerical one obtained by the Markov chain.

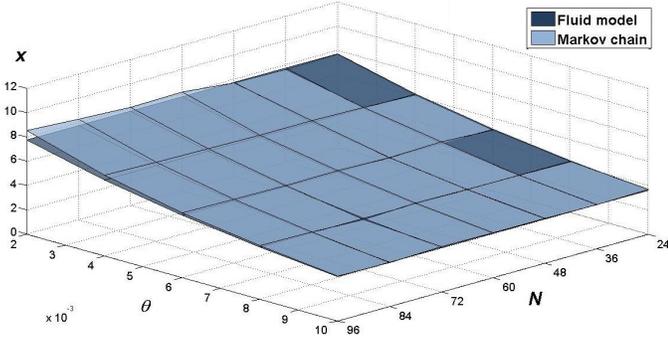


Fig. 11. Number of leeches in steady state, considering PWD scheme,  $\lambda=0.04$ ,  $\gamma=0.006$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $\mu_s=1.24$  and  $\alpha=1$

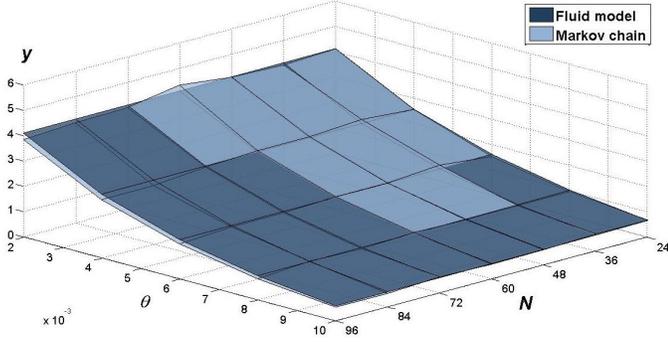


Fig. 12. Number of seeds in steady state, considering PWD scheme,  $\lambda=0.04$ ,  $\gamma=0.006$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $\mu_s=1.24$  and  $\alpha=1$

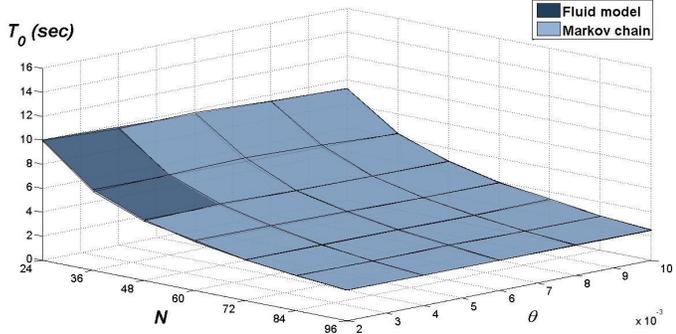


Fig. 13. Average time to download window 0 (initial delay), considering PWD scheme,  $\lambda=0.04$ ,  $\gamma=0.006$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $\mu_s=1.24$  and  $\alpha=1$

In Fig. 15 we evaluate the relation between the QoE parameters that are associated to initial delay and the design parameters  $\alpha$  and  $N$ . An analogous comparison is shown in Fig. 16 for the QoE parameters that are associated to interruptions. As expected, the higher the target MOS ( $M_{id,t}$  or  $M_{int,t}$ ), the higher the probability of no satisfaction,  $q_{id}$

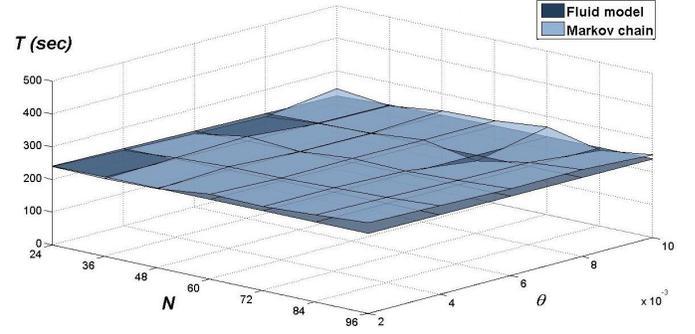


Fig. 14. Average time to download the whole file, considering PWD scheme,  $\lambda=0.04$ ,  $\gamma=0.006$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $\mu_s=1.24$  and  $\alpha=1$

and  $q_{int}$ , respectively. We can also appreciate that any of these probabilities increases when  $N$  diminishes, since a small  $N$  means a large window. In these figures it can also be appreciated the effect of  $\alpha$ : when it increases, the size of the initial window decreases and as a consequence  $q_{id}$  improves, but  $q_{int}$  is degraded.

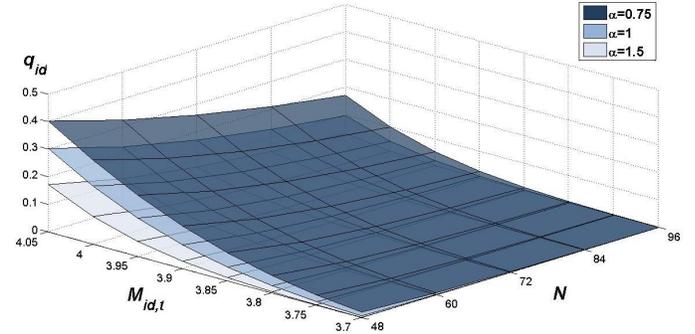


Fig. 15. Average time to download window 0 (initial delay), considering PWD scheme,  $\lambda=0.04$ ,  $\gamma=0.006$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $\mu_s=1.24$  and  $\alpha=1$

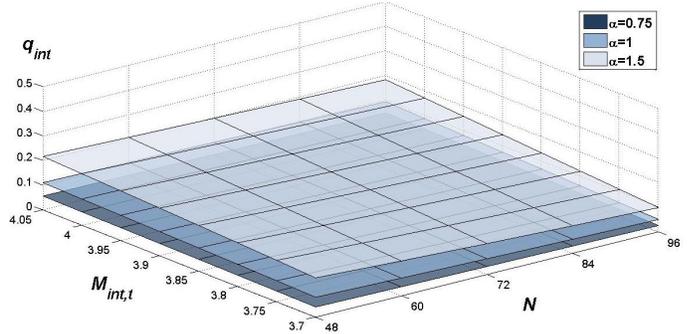


Fig. 16. Average time to download the whole file, considering PWD scheme,  $\lambda=0.04$ ,  $\gamma=0.006$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $\mu_s=1.24$  and  $\alpha=1$

In Tables II and III we summarize the values that were found for the design parameters  $\varepsilon$ ,  $N$ ,  $\alpha$  and  $\mu_s$  by following the proposed methodology described at the end of Section

$M_{id,t}$	$q_{id,t}$	$M_{int,t}$	$q_{int,t}$	$\varepsilon$	$N$	$\alpha$	$\mu_{s,pw}/\mu_{s,u}$
4.0	5%	4.0	5%	0.78	191	0.5	171.9 / 319.3
4.0	10%	4.0	10%	0.84	96	1	86.3 / 159.2
3.9	10%	4.0	10%	0.82	64	0.75	57.7 / 106.3
3.9	10%	3.9	10%	0.84	48	1	43.2 / 79.1

TABLE II

DESIGN PARAMETERS AS A FUNCTION OF THE QOE PARAMETERS SET FOR  $\lambda=0.04$ ,  $\theta=0.006$ ,  $\gamma=0.006$ ,  $c=0.00407$  AND  $\mu=0.00255$

$\theta$	$\gamma$	$\varepsilon$	$N$	$\alpha$	$\mu_{s,pw}/\mu_{s,u}$
0.006	0.006	0.82	64	0.75	43.2 / 79.1
0.006	0.02	0.69	64	0.75	72.3 / 123.8
0.01	0.006	0.65	48	1	37.0 / 62.4
0.01	0.02	0.58	48	1	54.2 / 92.2

TABLE III

DESIGN PARAMETERS AS A FUNCTION OF THE USERS' BEHAVIOR PARAMETERS SET FOR  $\lambda=0.04$ ,  $c=0.00407$ ,  $\mu=0.00255$ ,  $M_{id,t} = 3.9$ ,  $q_{id,t} = 10\%$ ,  $M_{int,t} = 4.0$  AND  $q_{int,t} = 10\%$

VII. Table II simply shows that when QoE targets are high, a large amount of server bandwidth must be provided. On the other hand, Table III confirms that a P2P network with cooperative seeds (small values of  $\gamma$ ) requires a small amount of extra server bandwidth, but also shows that when  $\theta$  is large, the required amount of extra server bandwidth is small, since the download demands are reduced, even though this situation implies the existence of non-cooperative leeches.

Finally, it must be remarked that in this work, we provide the necessary tools and analytical methodology to select the appropriate design parameters ( $\varepsilon$ ,  $N$  and  $\alpha$ ) in such a way that the expected QoE is satisfied, given that we know the basic system variables like the download/upload features of a video and a network, the users' behavior and a target set of QoE parameters; while considerably reducing the amount of server bandwidth that is needed to maintain the system in abundance conditions.

## IX. CONCLUSIONS AND FUTURE WORK

According to our analysis, in order to achieve abundance conditions in a P2P network, it is necessary the existence of cooperative peers. Since this cooperative scenario does not necessarily exist, we identify that in order to guarantee an acceptable performance of the system, servers that provide extra upload bandwidth must be installed. Since this extra bandwidth represents an additional cost, it is imperative to make an efficient use of it. Building on that, we proposed the PWD scheme. Our numerical evaluations showed that this scheme significantly reduces the amount of extra bandwidth required in the system to satisfy a set of QoE parameters, in comparison with a scheme that uniformly distributes those resources. We conclude that this scheme is one of our major contributions, since it implicitly takes advantage of the peers upload bandwidth, by assigning extra server bandwidth only to leeches that really need it, and, at the same time, making the system less dependent of peers behavior.

On the other hand, the window-based sharing mechanism not only provides an easy-to-implement and efficient way to interchange video files, but also allowed us to find a tradeoff between the initial delay and the interruption duration, by varying the size of the initial window. This is also an important contributions, since, to our knowledge, this tradeoff had not been previously analyzed in the context of QoE parameters (specifically, MOS).

In addition, we also developed an evaluation framework that can be used to calculate the design parameter of the system (number and sizes of windows and upload server bandwidth), that satisfy a set of target QoE parameters, given the behavior of the peers (peers arrival/ departure rates) and the network features (peers upload/download bandwidth). According to the numerical evaluations that we reported (which capture some of the features of YouTube service), we conclude that this framework is a powerful design tool that can be used by VoD servers providers in order to reduce their implementation costs, while controlling the QoE that they provide to their users.

Despite the previous conclusions, we have identified that future work must be done. Particularly, we are interested on including in our analysis the possibility of pause, forward or backward the video while it is being downloaded and shared, as well as considering varying peers upload/download bandwidth in our model.

## X. COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this paper.

## REFERENCES

- [1] Y. Huang, Y. Chen, *A Study on Content Delivery Strategy for QoE Enhancement in P2P IPTV*, IEEE International Conference on Computer and Information Technology (CIT), Sept. 2014, pp. 520-527.
- [2] D. Kondo, et al, *P2P live streaming system for multi-view video with fast switching*, 16th International Telecommunications Network Strategy and Planning Symposium (Networks), Sept. 2014, pp. 1-7.
- [3] O. Ojo, A. Oluwatope, O. Ogunsoola, *UStream: Ultra-Metric Spanning Overlay Topology for Peer-to-Peer Streaming Systems*, IEEE International Symposium on Multimedia (ISM), Dec. 2015, pp. 601-604.
- [4] Y. Zhou, T. Z. J. Fu, D. M. Chiu, *A Unifying Model and Analysis of P2P VoD Replication and Scheduling*, IEEE/ACM Transactions on Networking, vol. 23, no. 4, Aug. 2015, pp. 1163-1175
- [5] P. Romero, F. Robledo, P. Rodriguez-Boca, C. Rostagnol, *Analysis and Design of Peer-assisted Video-on-Demand Services*, International Transactions in Operational Research, vol. 21, no. 4, July 2014, pp. 559-579.
- [6] W. Wu, R. T. B. Ma, J. C. S. Lui, *Distributed Caching via Rewarding: An Incentive Scheme Design in P2P-VoD Systems*, IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 3, March 2014, pp. 612 - 621.
- [7] M. Faiqurahman, A. I. Kistijantoro, *Implementation of modified Probabilistic Caching Schema on Bittorrent protocol for video on demand content*, International Seminar on Intelligent Technology and Its Applications (ISITIA), May 2015, pp. 357-362.
- [8] M. Wichtlhuber, et al, *QTrade: a quality of experience based peer-casting trading scheme*, IEEE International Conference on Peer-to-Peer Computing (P2P), Sept. 2015, pp. 1-10.

- [9] T. Rohmer, A. Nakib, A. Nafaa, *Priori Knowledge Guided Approach for Optimal Peer Selection in P2P VoD Systems*, IEEE Transactions on Network and Service Management, vol. 11, no. 3, Sept. 2014, pp 350 - 362.
- [10] R. Dubin, et al, *Hybrid Clustered Peer-Assisted DASH-SVC System*, IEEE International Conference on CIT/UCC/DAS/PICOM, 2015, pp 1651- 1656.
- [11] Ciullo, et al, *Peer-Assisted VoD Systems: An Efficient Modeling Framework*, IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 7, July 2014, pp 1852 - 1863.
- [12] Global Internet Phenomena. Latin America and North America Report 2015, [<https://www.sandvine.com>].
- [13] Ericsson Mobility Report June 2015, [<https://www.ericsson.com/mobility-report>].
- [14] S. Jia, et al, *Modelling of P2P-Based Video Sharing Performance for Content-Oriented*, Mobile Information Systems, Vol. 2016, Article ID 1319497, 13 pages, [<http://dx.doi.org/10.1155/2016/1319497>].
- [15] C. Xu, S. Jia, L. Shong and G. Muntean, *Socially Aware Mobile Peer-to-Peer Communications for Community Multimedia Streaming Services*, IEEE Communications Magazine, vol. 53, no. 10, Oct. 2015, pp 150-156.
- [16] E. Baez-Esquivel, M. E. Rivero-Angeles and G. Rubino, *Priority scheme for window-based video-on-demand transmission on BitTorrent-like peer-to-peer networks*, Proc. IEEE International Conference on Communications (ICIC), Jun. 2013, pp. 3000-3005.
- [17] M. E. Rivero-Angeles and G. Rubino, *Priority-Based Scheme for File Distribution in Peer-to-Peer Networks*, Proc. IEEE International Conference on Communications (ICIC), May 2010, pp. 1-6.
- [18] S. Tewari, L. Kleinrock, *Analytical Model for BitTorrent-Based Live Video Streaming*, 4th IEEE Consumer Communications and Networking Conference, Jan. 2007, pp. 976-980.
- [19] D. Qiu and R. Srikant, *Modeling and performance analysis of BitTorrent-like peer-to-peer networks*, Conference on Applications, technologies, architectures, and protocols for computer communications, Oct. 2004.
- [20] C. Liang, Z. Fu, Y. Liu, C. W. Wu, *Incentivized Peer-Assisted Streaming for On-Demand Services*, IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 9, Sept. 2010.
- [21] T. Hossfeld, et al, *Initial delay vs. interruptions: between the devil and the deep blue sea*, IEEE Fourth International Workshop on Quality of Multimedia Experience (QoMEX), Jul. 2012, pp 1-6.
- [22] J. J. Ramos-Munoz, et al, *Characteristics of mobile Youtube traffic*, IEEE Wireless Communications, vol. 21, no. 1, Feb. 2014, pp. 18-25.
- [23] G. Dimopoulos, P. Barlet-Ros, J. Sanjuas-Cuxart, *Analysis of YouTube user experience from passive measurements*, IEEE 9th International Conference on Network and Service Management (CNSM), Oct. 2013, pp. 260-267.
- [24] P. T. Anh-Quang, K. Piamrat, C. Viho, *QoE-aware routing for video streaming over ad-hoc networks*, IEEE Global Communications Conference, Dec. 2014, pp. 181-186.