Hindawi
Mobile Information Systems
Volume 2017, Article ID 2084684, 18 pages
https://doi.org/10.1155/2017/2084684

Research Article

A Window-Based, Server-Assisted P2P Network for VoD Services with QoE Guarantees

Noé Torres-Cruz,[1,2] Mario E. Rivero-Angeles,[1] Gerardo Rubino,[3] Ricardo Menchaca-Mendez,[1] and Rolando Menchaca-Mendez[1]

[1] *Centro de Investigación en Computación, Instituto Politécnico Nacional, Av. Juan de Dios Bátiz, Esq. Miguel, Othón de Mendizábal, Col. Nueva Industrial Vallejo, 07738 Ciudad de México, Mexico*
[2] *UPIITA, Instituto Politécnico Nacional, Av. IPN, No. 2580, Col. Barrio la Laguna Ticomán, 07340 Ciudad de México, Mexico*
[3] *INRIA Rennes, Bretagne Atlantique, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France*

Correspondence should be addressed to Noé Torres-Cruz; ntorresc@ipn.mx

Received 29 October 2016; Revised 19 January 2017; Accepted 1 February 2017; Published 23 March 2017

Academic Editor: Ben Niu

We describe a Peer-to-Peer (P2P) network that is designed to support Video on Demand (VoD) services. This network is based on a video-file sharing mechanism that classifies peers according to the window (segment of the file) that they are downloading. This classification easily allows identifying peers that are able to share windows among them, so one of our major contributions is the definition of a mechanism that could be implemented to efficiently distribute video content in future 5G networks. Considering that cooperation among peers can be insufficient to guarantee an appropriate system performance, we also propose that this network must be assisted by upload bandwidth from servers; since these resources represent an extra cost to the service provider, especially in mobile networks, we complement our work by defining a scheme that efficiently allocates them only to those peers that are in windows with resources scarcity (we called it *prioritized windows distribution scheme*). On the basis of a fluid model and a Markov chain, we also developed a methodology that allows us to select the system parameters values (e.g., windows sizes or minimum servers upload bandwidth) that satisfy a set of Quality of Experience (QoE) parameters.

1. Introduction

Per-to-Peer (P2P) networks have been widely used to increase the capacity of systems due to the fact that nodes cooperate among them to reduce 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 supposed to be used only after their download completion. However, in recent years a large body of research has been focused on analyzing these networks for video distribution. In these services it is necessary to consider that the video playback is initiated even if its download is still in progress. Services like Live IPTV and Video on Demand (VoD) have been considered in the context of P2P networks in [1–11].

VoD streaming has become widely popular because of its particular features: subscribers are allowed to select and play back a video as well as to rewind, fast-forward, pause, or stop it; and the playback startup time (also known as initial delay) is considerably reduced since the playback can be initiated even if the video download is not completed. These features have made VoD services very attractive and nowadays represent an important proportion of the current Internet traffic. According to [12], VoD represented approximately 23% of the total traffic in mobile networks in North America, during 2015; 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]. In this paper, we analyze a P2P network for VoD services by means of both a fluid model and Markov chains. These analytic tools capture the main characteristics of the distribution of a video file among the nodes in a P2P network. Building on this, we obtain different performance metrics that accurately describe the quality of the video streaming service as experienced by a user in terms of initial delay and duration of the pauses along the playback. These parameters are considered in order to obtain a Quality of Experience (QoE) score as detailed further in the paper.

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 a video file into many pieces called *chunks*. The peers download a specific video file by exchanging the corresponding chunks according to 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 to download 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 specific download order 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 [16–18] is an example of such modifications.

The analyzed scenario in [16–18] consists of a group of peers that have required the same video-file download in a VoD-P2P system. In order to efficiently distribute this file, the authors proposed the following procedure. The set of ordered chunks that composes the shared video file is divided into N segments of equal size. These segments are called *windows*; they are denoted by $w_0, w_1, \ldots, w_{N-1}$; and they follow the chunks' order in the stream, as it is shown in Figure 1. The basic idea is that peers must be classified in the system according to the window they are currently downloading and this information must be dynamically updated in the tracker. As a consequence, all leeches can 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 that are downloading the file at window w_j, for $j > i$. Conversely, the peers that are downloading the file in window w_i can serve any other peer that is downloading the file at window w_j, for $j < i$. By enabling this peer selection strategy, leeches only have to know the current window of the other peers, rather than the chunks that each peer possesses. This facilitates the task of identifying the peers to connect to.

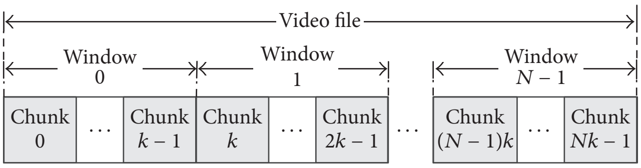


Figure 1: Video file divided into windows.

For this window-based strategy, it is assumed that (a) leeches begin the file downloading process at window zero; (b) leeches at window w_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.

In order to guarantee an adequate service in terms of a QoE score, we propose a number of key modifications to the previously described window-based strategy. Specifically, unlike the works presented in [16–18], where windows were assumed to have the same size, here we propose that the size of window 0 can be different from the size of the remaining windows. The rationale behind this modification is to achieve a trade-off between the initial delay and the pauses duration.

We also propose a methodology to select the size of the windows, by developing a mathematical analysis that allows us to calculate the probability distribution of both, the initial delay and the pauses duration.

Through numerical results, we evaluate the analytical expressions derived in this work and we conclude that in order to guarantee an acceptable level of 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 cannot guarantee an acceptable performance under some conditions that depend on the random nature of arrivals and departures of peers and the level of cooperation of the peers. Specifically, we calculate the exact amount of additional bandwidth required to achieve conditions that guarantee that all leeches in the system are able to download the file at the maximum rate. This is another major contribution of this paper.

Evidently, the utilization of this additional bandwidth increases the implementation cost of the system; therefore, we propose a novel chunk distribution scheme, named prioritized windows distribution (PWD), where the servers provide more bandwidth to leeches in higher windows, rather than a uniform distribution as it was originally proposed. In the uniform distribution scheme, leeches in lower windows are served by seeds and other leeches in higher windows, while leeches in higher windows are only served by seeds and a small amount of leeches. For that reason, a large amount of resources from servers must be provided to leeches in higher windows. The rationale behind PWD is to counteract this disadvantage.

Lastly, we provide strict guidelines on how to select the system parameters such that QoE guarantees can be provided under different system conditions.

The rest of the paper is organized as follows: Section 2 discusses some of the previous work in the area and makes a detailed comparison with our proposal. Then, Section 3 presents in detail the window-based system, including the main assumptions and considerations. Section 4 explains the server-assisted P2P network proposal in order to attain abundance conditions in the system, including the prioritized windows distribution scheme to reduce the amount of extra server capacity. Following this, in Section 5 we derive the probability distributions of initial delay and interruption duration. Then, the implications of the window-based scheme on the QoE level are discussed in Section 6. 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.

2. 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 [19]. 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 topologies. For instance, in [3] a hybrid tree-mesh topology is proposed. Other researchers have identified that a way to increase the capacity of a P2P network is by implementing strategies that efficiently distribute content (videos) among the population of peers. Examples of this kind of work are [4–6]. Schemes that incentivize cooperation among peers to increase the upload network capacity have been proposed on works such as [6–8, 20]. Finally, the main focus of some other research works 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 works are [7, 9, 10, 14, 15, 20]. The scheme considered in this paper belongs to the last class, because the proposed window-based scheme finds the best peers 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 proposal has not been published so far.

On the other hand, different analytical tools have been used to model P2P networks, including fluid models. In [21] 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 does not consider the specific features of VoD services. In [16–18] 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 analyze a windows-based P2P network; but, in order to guarantee an adequate level of service, we additionally consider that the system 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 models, 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 assumption 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 design a sharing mechanism that is significantly simpler than the one described in [7].

In [20], the authors present a system which achieves scheduling video sharing between peers by adopting a dynamic buffering-progress-based scheme: a downloader receives chunks only from peers with a similar playback progress. Though this proposal has some similarities with our window-based one (the chunk sharing mechanism is based on the download progress of peers), and though in both works a fluid model is used, the analysis perspective is quite different. For instance, they propose that leeches stay in the system until the downloading is finished, while we use a more realistic model in which leeches can leave the system at certain rate (denoted by θ). 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 (our PWD scheme), rather than incentivizing cooperation, as they suggest. Note that their cooperation scheme can also be applied to our system in order to further reduce the assisted server bandwidth.

In [11], the authors propose a modeling framework to compute the required server bandwidth, which has several similarities with our modeling work: a Poisson arrival process, analysis restricted to only one video, and homogeneous download rate. Though the model that they propose considers a wider range of scenarios than ours (nonstationary traffic, heterogeneous upload rate), our contribution is not 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 do not degrade significantly the users' experience (measure through QoE parameters); the integration of such a consideration, as well as its analysis, is another contribution of this paper, since none of the above described works has addressed it and, to our knowledge, little research has been done about this kind of issue. In order to consider the effect of QoE parameters, we use the experimental results reported in [22], where relations between QoS parameters (initial delays and interruptions duration) and QoE parameters (Medium Opinion Score, widely known as MOS) are established.

3. The Model for the Window-Based System

In this section we present the mathematical model for the *conventional* window-based strategy. As we mentioned in Section 1, this strategy was proposed for chunk distribution among peers in a P2P-VoD system. We refer to this system as *conventional* since it relies only on the bandwidth shared by the peers in the network as proposed in [16, 21]. 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 it can only rely on the cooperation among users. This section also presents the main assumptions and parameters of the system.

In order to facilitate the reading of Sections 3–6, we summarize in Most Relevant Variables Summary section the most relevant variables that are used through the whole paper.

The number of leeches that are downloading 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,

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

There is a single file in the system, assumed for simplicity to be of size 1 as in [21]. 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 its chunks to any other leech currently downloading data in 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. We also assume that new leeches arrive at window 0 with constant rate λ. Observing that we are not considering Poisson processes here, the model is a deterministic one. The transition rate from any peer at window i to the next one is denoted by τ_i and the rate at which a leech leaves the system before the completion of the download or the playback process is denoted by θ. A seed leaves the system with rate γ. We assume that all peers have the same physical characteristics. Specifically, they all have the same uploading bandwidth μ (in files/sec). We denote by $1/c$ the (mean) time needed to download the whole file without interruptions working at full capacity; hence, c is the maximal file 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, $\mu_w = N\mu$. Recall that the file size is 1.

In this analysis, we consider that peers have complete knowledge of the state of system, namely, that peers know the current group of all the other peers. This is a reasonable assumption, since, in a BitTorrent-like system, peers can obtain this information from the tracker server. Under this assumption, if the number of leeches and seeds is sufficiently high, all leeches download a window at the maximum rate c_w. This condition is referred to as *abundance*. However, when there are not enough peers in the system, the leeches download a window at a rate that is smaller than c_w. 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, that is, 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, $x_i(t)$, and of the number of seeds, $y(t)$, for the system satisfies the following equations:

$$\begin{aligned} x_0'(t) &= \lambda - \theta x_0(t) - \tau_0, \\ x_i'(t) &= \tau_{i-1} - \theta x_i(t) - \tau_i, \quad 1 \leq i \leq N-1, \\ y'(t) &= \tau_{N-1} - \gamma y(t), \end{aligned} \tag{2}$$

where

$$\tau_i = \min\left\{ c_w x_i(t), \mu_w x_i(t) \cdot \left(\sum_{k=i+1}^{N-1} \frac{x_k(t)}{\sum_{j=0}^{k-1} x_j(t)} + \frac{y(t)}{x(t)} \right) \right\}, \tag{3}$$

for $i = 0, 1, \ldots, N-1$.

In the case of the last window, that is, when $i = N-1$, we have

$$\tau_{N-1} = \min\left\{ c_w x_{N-1}(t), \mu_w x_{N-1}(t) \frac{y(t)}{x(t)} \right\}. \tag{4}$$

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 a window at a rate which is described as follows. First of all, note that every peer can upload a window at rate μ_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 $x_i(t)/x(t)$. 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 $x_i(t)/\sum_{j=0}^{k-1} x_j(t)$. Leeches in the last window only receive chunks from seeds.

Let us compute the equilibrium point of this dynamical system, which we denote by $(\overline{x}_0, \overline{x}_1, \ldots, \overline{y})$. This equilibrium point can be obtained by solving the system $\{x_0'(t) = 0, x_1'(t) = 0, \ldots, y'(t) = 0\}$ and by replacing $x_i(t)$ by $\overline{x}_i$ and $y(t)$ by $\overline{y}$ in (2); hence, we obtain a system of equations given by

$$\begin{aligned} \lambda - \theta\overline{x}_0 - \tau_0 &= 0, \\ \tau_{i-1} - \theta\overline{x}_i - \tau_i &= 0, \quad 1 \le i \le N-1, \\ \tau_{N-1} - \gamma\overline{y} &= 0. \end{aligned} \tag{5}$$

Though this system can be solved in abundance or in penury (see (3) and (4)), we focus our analysis in the former one, because a target QoE score can be guaranteed if this abundance condition exists, as we show later in this paper. Considering this, we substitute $\tau_i = c_w\overline{x}_i$ in (5) and by solving the system we obtain

$$\begin{aligned} \overline{x}_i &= \frac{\lambda c_w^i}{(\theta + c_w)^{i+1}}, \\ \overline{y} &= \frac{\lambda}{\gamma}\left(\frac{c_w}{\theta + c_w}\right)^N, \end{aligned} \tag{6}$$

for $0 \le i \le N-1$.

The total number of leeches in equilibrium, $\overline{x}$, is given in the next equation:

$$\overline{x} = \sum_{i=0}^{N-1} \overline{x}_i = \frac{\lambda}{\theta}\left[1 - \left(\frac{c_w}{\theta + c_w}\right)^N\right]. \tag{7}$$

All numerical explorations showed convergence towards equilibrium (see the numerical results that are shown in Section 8). The analysis of the stability of this nonlinear system of differential equations is left for future work.

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 (γ) is an important variable, considering that seeds already have all the chunks of the video file. From (3), it can be observed that in order to have abundance at window w_i, the following condition must be met:

$$c_w x_i(t) \le \mu_w\left(\sum_{k=i+1}^{N-1} x_k(t)\frac{x_i(t)}{\sum_{j=0}^{k-1} x_j(t)} + y(t)\frac{x_i(t)}{x(t)}\right). \tag{8}$$

If we consider $t = \infty$ in (8), we can substitute $x_i(t)$ and $y(t)$ by the definitions of $\overline{x}_i$ and $\overline{y}$ that were given in (6); in other words, we are analyzing the system at the equilibrium point. Hence, the condition that must be met in order to have abundance at window w_i can also be expressed as

$$\begin{aligned} c_w \le \mu_w \Bigg(&\sum_{k=i+1}^{N-1} \frac{c_w^k/(\theta + c_w)^{k+1}}{\sum_{j=0}^{k-1}\left(c_w^j/(\theta + c_w)^{j+1}\right)} \\ &+ \frac{\theta\left(c_w/(\theta + c_w)\right)^N}{\gamma\left[1 - \left(c_w/(\theta + c_w)\right)^N\right]}\Bigg). \end{aligned} \tag{9}$$

Notice that in (9) the abundance condition is given as a function of the system parameters; consequently, by simplifying this expression and by clearing γ, we can find a value γ_i that represents the maximum seeds departure rate that guarantees abundance in window i (i.e., $\gamma \le \gamma_i$, for all i).

$$\begin{aligned} &\gamma_i \\ &= \frac{(\theta + c_w)c_w^N/\left((\theta + c_w)^N - c_w^N\right)}{c_w(\theta + c_w)/\theta\mu_w - \sum_{k=i+1}^{N-1}\left(c_w^k/\left((\theta + c_w)^k - c_w^k\right)\right)}. \end{aligned} \tag{10}$$

Rate γ is clearly a *manager* variable that could be controlled according to (10). In Figure 2, the abundance condition for different values of the departure rate for a leech (θ) 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; that is, the seeds 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 to download windows at their maximum capacity. In other words, our analysis allows finding an appropriate value for this control variable. However, achieving the adequate value of γ, that is, 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 of coping with this problem is to introduce penalties or rewards to peers in order to encourage a cooperative behavior (as in [6–8, 16, 18, 20]). However, these mechanisms do not guarantee to achieve an acceptable QoE in the network since they still relay 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.

From (10) it can also be observed that abundance condition could be achieved by controlling N, the number of windows. Figure 3 shows that, by reducing N, the required value of γ_{N-1} to guarantee abundance is slightly relaxed (increased); however, reducing N deteriorates the QoE parameters, as it is widely discussed in Sections 5–7 of this paper.

The rest of the variables that are involved in (10) are much harder to control, since the download and upload bandwidths (c_w and μ_w, resp.) are usually fixed by the hardware used and

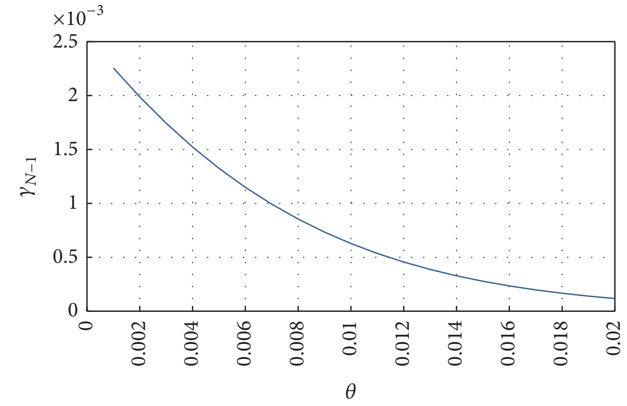


Figure 2: Relationship between minimal departure rate for a seed and departure rate for leeches in order to keep the system in abundance conditions; $\lambda = 0.04$, $c = 0.00407$, $\mu = 0.00255$, and $N = 48$.

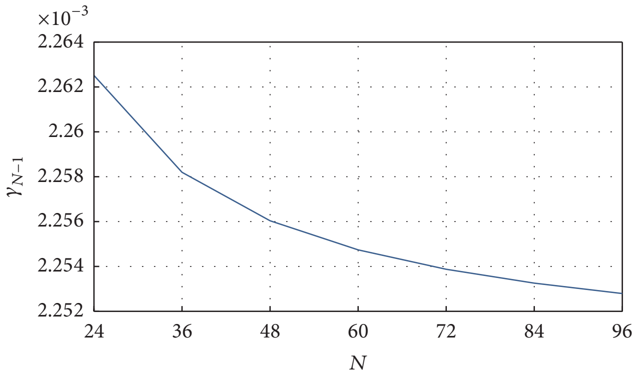


Figure 3: Relationship between minimal departure rate for a seed and number of windows in order to keep the system in abundance conditions; $\lambda = 0.04$, $c = 0.00407$, $\mu = 0.00255$, and $\theta = 0.001$.

the arrival rate (λ) depends on the file popularity, that is, 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 to use a different size for the first window in order to achieve a trade-off between the initial playback delay and the pauses duration. Second, we propose the use of additional server bandwidth that the network manager can explicitly provide to satisfy QoE parameters. Third, we propose an efficient distribution scheme of these extra servers resources, called *prioritized windows distribution*, which is based on the window of the leech that is downloading the file. These mechanisms can be fully controlled by the network manager and are explained in detail in the following sections.

4. Server-Assisted System and Reduced Initial Window

According to [22] the QoE that a user experiences during an online video playback is highly determined by the initial delay, by probability that a pause occurs during the playback, and by the duration of such pauses. In this regard, it is important to note that there is a trade-off between the values that must be assigned to these parameters, 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 downloading the video file for a long period of time before the playback begins, a larger portion of that file will be available for its future playback; hence, the probability that a pause occurs is reduced.

Among these parameters, it has been identified that the pause probability and the duration of these pauses are much more harmful to the user's perception than the initial delay. In general, users prefer to wait longer times at the beginning of the file playback, if this means that the video will not be 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 the windows in the system since the bandwidth required to attain abundance conditions is much harder with a large number of windows, as explained later in this section. However, the size of the first window can be modified in order to guarantee an acceptable initial delay, while the rest of the windows can be set to an adequate value.

Considering the previously described trade-off, we propose the following modifications to the conventional window-based system in order to provide an acceptable level of QoE:

(i) Unlike the conventional model described in the previous section, here we assume that the size of the first window can be different from the size of the remaining windows. This is important because these window sizes define a trade-off between initial delay and average pause duration.

(ii) In order to prevent unacceptable levels of initial delay or average pause duration, we propose to enhance the P2P network with a fixed download bandwidth provided by servers.

(iii) Lastly, a nonuniform chunk distribution scheme is proposed in order to avoid the resource starving problem experienced by leeches in upper windows. This scheme effectively reduces the required server bandwidth to reach abundance conditions.

Taking into account the previous considerations, we denote by μ_s the bandwidth provided by servers, and as an initial analysis we assume that this extra bandwidth is uniformly distributed among all the leeches, that is, in the same way that the bandwidth provided by peers is distributed (as previously mentioned, in Section 4.2 we propose a more efficient bandwidth distribution scheme); 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. As a consequence, the average 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 α, which represents the ratio of the size of window i $(i > 0)$ to the size of window 0; hence $\alpha = d_1/d_0$. Since the average download time for the whole file must be equal to the sum of the average download times of all the windows, that is, $1/c = d_0 + (N+1)d_1$, we can express c_0 and c_1 as follows:

$$\begin{aligned} c_0 &= c\left(\alpha N + 1 - \alpha\right), \\ c_1 &= c\left(N + \frac{1-\alpha}{\alpha}\right). \end{aligned} \tag{11}$$

Following an analogous analysis to the previous paragraph, the upload rates for windows 0 and i are, respectively, given by the following expressions:

$$\begin{aligned} \mu_0 &= \mu\left(\alpha N + 1 - \alpha\right), \\ \mu_1 &= \mu\left(N + \frac{1-\alpha}{\alpha}\right). \end{aligned} \tag{12}$$

The incorporation of parameters μ_s, c_0, c_1, μ_0, and μ_1 in our analysis does not modify the essence of the fluid model described in Section 3. However, the definition of τ_i must be modified since in this case the rates to download or upload a window are given by (11)-(12) and, additionally, the upload capacity of the system must be increased by the server bandwidth. As a result of these new considerations, we have the following expressions (where time t is omitted for the sake of room):

$$\begin{aligned} &\tau_0 \\ &\quad = \min\left\{c_0 x_0, x_0\left[\mu_0\left(\sum_{k=1}^{N-1}\frac{x_k}{\sum_{j=0}^{k-1}x_j} + \frac{y}{x}\right) + \frac{\mu_s}{x}\right]\right\}, \\ &\tau_i \\ &\quad = \min\left\{c_1 x_i, x_i\left[\mu_1\left(\sum_{k=i+1}^{N-1}\frac{x_k}{\sum_{j=0}^{k-1}x_j} + \frac{y}{x}\right) + \frac{\mu_s}{x}\right]\right\}, \end{aligned} \tag{13}$$

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

$$\tau_{N-1} = \min\left\{c_w x_{N-1}, x_{N-1}\frac{\mu_1 y + \mu_s}{x}\right\}. \tag{14}$$

Again, working as in Section 3, the new equilibrium point of the system in abundance is now given by

$$\begin{aligned} \overline{x}_0 &= \frac{\lambda}{\theta + c_0}, \\ \overline{x}_i &= \frac{\lambda c_0}{(\theta + c_0)}\frac{c_1^{i-1}}{(\theta + c_1)^i}, \end{aligned} \tag{15}$$

for $1 \le i \le N-1$, and

$$\overline{y} = \frac{\lambda}{\gamma}\frac{\alpha}{\beta}\left(\frac{c_1}{\theta + c_1}\right)^N, \tag{16}$$

where $\beta = (\theta + c_0)/(\theta + c_1)$. The total number of leeches in equilibrium, $\overline{x}$, is given by the following expression:

$$\overline{x} = \frac{\lambda}{\theta}\left(1 - \frac{\alpha}{\beta}\left(\frac{c_1}{\theta + c_1}\right)^N\right). \tag{17}$$

It is important to note that even if the parameters related to the video-file uploading (μ_0, μ_1, and μ_s) are not explicit in previous equations, their values are fundamental to guarantee the abundance condition of the system, as it will be shown in the next subsection. Finally, it must be said that (13)–(17) are reduced to their corresponding counterparts given by (3)–(7), when $\alpha = 1$ (i.e., the sizes of all windows are equal).

4.1. Minimum Server Bandwidth Requirement to Guarantee Abundance Conditions. According to (13)–(17), the abundance condition can be expressed by

$$c_0 \le \mu_0\left(\sum_{k=1}^{N-1}\frac{x_k}{\sum_{j=0}^{k-1}x_j} + \frac{y}{x}\right) + \frac{\mu_s}{x} \tag{18}$$

for window 0, by

$$c_1 \le \mu_1\left(\sum_{k=i+1}^{N-1}\frac{x_k}{\sum_{j=0}^{k-1}x_j} + \frac{y}{x}\right) + \frac{\mu_s}{x} \tag{19}$$

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

$$c_1 \le \mu_1\frac{y}{x} + \frac{\mu_s}{x} \tag{20}$$

for window $N-1$.

From (18) to (20), the abundance conditions in terms of γ can be found as shown in Section 3. However, since γ is a parameter that highly depends on users' behavior, a complementary way to guarantee abundance in the system can be based on μ_s. As it was previously mentioned, we propose different schemes in order to guarantee abundance with a minimum value of μ_s (nonuniform 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 γ 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 $\mu_s \ge \mu_{\min}^i$, for every i. From (18) to (20) we obtain

$$\begin{aligned} \mu_{\min}^0 &= \max\left\{0, x\left[c_0 - \mu_0\left(\sum_{k=1}^{N-1}\frac{x_k}{\sum_{j=0}^{k-1}x_j} + \frac{y}{x}\right)\right]\right\}, \\ \mu_{\min}^i &= \max\left\{0, x\left[c_1 - \mu_1\left(\sum_{k=i+1}^{N-1}\frac{x_k}{\sum_{j=0}^{k-1}x_j} + \frac{y}{x}\right)\right]\right\} \end{aligned} \tag{21}$$

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

$$\mu_{\min}^{N-1} = \max\left\{0, c_1 x - \mu_1 y\right\}. \tag{22}$$

In order to understand the intuition behind the max operation in (21)-(22), 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, we can write

$$\mu_{\min}^{i-1} = \max\left\{0, \mu_{\min}^{i} - \frac{\mu_1 x x_i}{\sum_{j=0}^{i-1} x_j}\right\} \tag{23}$$

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

Since x, x_i, and μ_1 are always nonnegative values, it is clear that $\mu_{\min}^{i} \ge \mu_{\min}^{i-1}$ for $2 \le i \le N-1$. 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; that is, the bandwidth required from servers to guarantee abundance in the whole system is $\mu_{\min} = \mu_{\min}^{N-1}$.

After substituting (16) and (17) in (22) we have that

$$\mu_{\min} = \max\left\{0, \frac{\lambda c_1}{\theta}\left[1 - \frac{\alpha}{\beta}\left(1 + \frac{\theta \mu_1}{\gamma c_1}\right)\left(\frac{c_1}{\theta + c_1}\right)^{N}\right]\right\}. \tag{24}$$

It is important to remark that if $\alpha = 1$, then the model of this section is reduced to the conventional window-based model; consequently, x_i and y are simply given by (6).

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 (23), 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 Figure 4, where the amount of upload bandwidth that a leech in window i can get (UB_i) 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 next subsection.

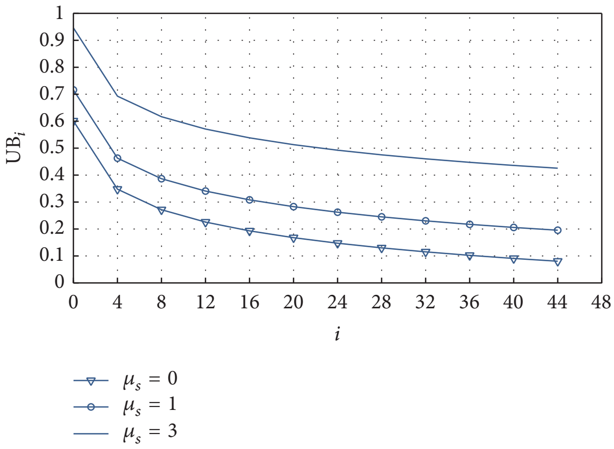


Figure 4: Upload bandwidth per peer in window i, for $\lambda = 0.04$, $c = 0.00407$, $\mu = 0.00255$, $\theta = 0.001$, $\gamma = 0.006$, $\alpha = 1$, and $N = 48$.

4.2. 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, the server bandwidth assigned to leeches in window i will now 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 to as *prioritized windows distribution* (PWD) scheme, while the one described in Section 4.1 is referred to as *uniform distribution* (UD) scheme.

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

$$\begin{aligned} \tau_0 &= \min\left\{c_0 x_0, \right. \\ &\qquad \left. x_0\left[\mu_0\left(\sum_{k=1}^{N-1}\frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x}\right) + \frac{\mu_s}{x_\varepsilon}\right]\right\}, \\ \tau_i &= \min\left\{c_1 x_i, \right. \\ &\qquad \left. x_i\left[\mu_1\left(\sum_{k=i+1}^{N-1}\frac{x_k}{\sum_{j=0}^{k-1} x_j} + \frac{y}{x}\right) + \frac{(i+1)^{\varepsilon}\mu_s}{x_\varepsilon}\right]\right\} \end{aligned} \tag{25}$$

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

$$\tau_{N-1} = \min\left\{c_w x_{N-1}, x_{N-1}\left[\frac{\mu_1 y}{x} + \frac{N^{\varepsilon}\mu_s}{x_\varepsilon}\right]\right\}, \tag{26}$$

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 μ_s.

From (25) to (26), it can be observed that the priority of window i over window $i-1$ (for $1 \le i \le N-1$) is more accentuated, if ε increases. If $\varepsilon = 0$, these equations are reduced to (13)-(14). 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 modify the uplink capacity of the system, in the way that was already considered in (25)-(26).

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

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

$$\mu_i = \max\left\{0,\ \frac{x_\varepsilon}{(i+1)^\varepsilon}\left[c_1 - \mu_1\left(\sum_{k=i+1}^{N-1}\frac{x_k}{\sum_{j=0}^{k-1}x_j} + \frac{y}{x}\right)\right]\right\} \quad (27)$$

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

$$\mu_{N-1} = \max\left\{0, \frac{x_\varepsilon}{N^\varepsilon}\left[c_1 - \mu_1\frac{y}{x}\right]\right\}. \quad (28)$$

It is important to emphasize that if ε 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 servers must be installed to guarantee abundance in those windows. Hence, ε 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 Figure 5 we show $\mu^i_{\min}$ for different values of ε. 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 ε (ε_{opt}). However, we can find this optimal value by numerically evaluating the following expressions:

$$i_{\text{crit}}(\varepsilon) = \underset{i}{\operatorname{argmax}}\left\{\mu_i(\varepsilon)\right\}, \quad (29)$$

$$\varepsilon_{\text{opt}} = \underset{\varepsilon}{\operatorname{argmin}}\left\{\mu_{i_{\text{crit}(\varepsilon)}}\right\}. \quad (30)$$

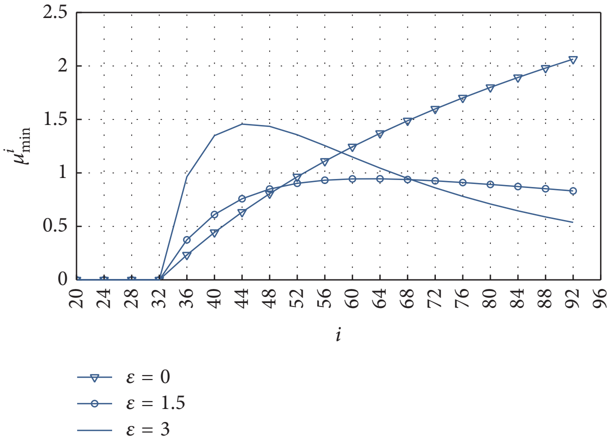


Figure 5: Minimum server bandwidth required by window i for $\lambda = 0.04$, $c = 0.00407$, $\mu = 0.00255$, $\theta = 0.001$, $\gamma = 0.006$, and $N = 96$.

Here, $i_{\text{crit}}(\varepsilon)$ represents the index of the window that requires the largest value of $\mu_i^{\min}$ for a given ε. Strictly, ε_{opt} depends on all the parameters of the system; however, after evaluating (29) and (30), we found that it only significantly depends on θ and γ, as shown in Figure 6. Considering this, we propose to calculate ε_{opt} through the approximation given by

$$\varepsilon_{\text{opt}} \approx 4916\theta^2 - 139\theta + 2279\theta\gamma - 49\gamma + 971\gamma^2 + 1.65. \quad (31)$$

The polynomial coefficients in (31) were obtained by applying a second-order linear regression to the results shown in Figure 6. In Figure 7 we show a comparison between the exact evaluation and the corresponding approximation. Lastly, it must be noticed that even if ε is chosen exclusively in terms of θ and γ, $\mu_{\min}$ is still a function of the remaining parameters of the system (e.g., N and α). In Section 8, we show that using the evaluation of ε_{opt} that is defined by (31), the PWD scheme significantly reduces $\mu_{\min}$ in comparison with the UD scheme.

5. 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 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. For this purpose, we change the viewpoint and interpret now the system probabilistically. Specifically, in this section we model the probability distributions of the initial delay and the interruption duration along the playback of a video which is being downloaded from the described system.

Since the mathematical analysis in the previous sections is valid only if the abundance condition exists, the analysis in this section is also limited to such circumstances; that is,

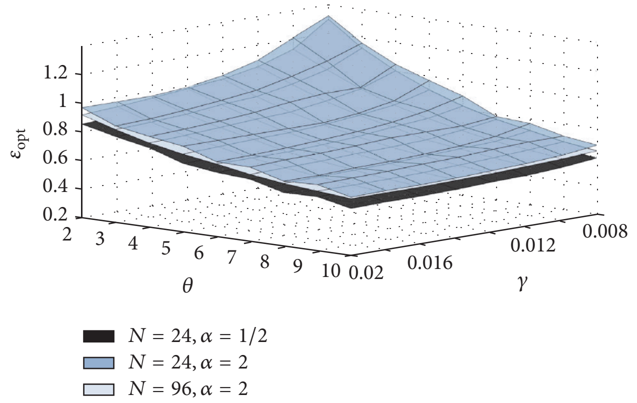


Figure 6: Optimal values of ε for $\lambda = 0.04$, $c = 0.00407$, and $\mu = 0.00255$.

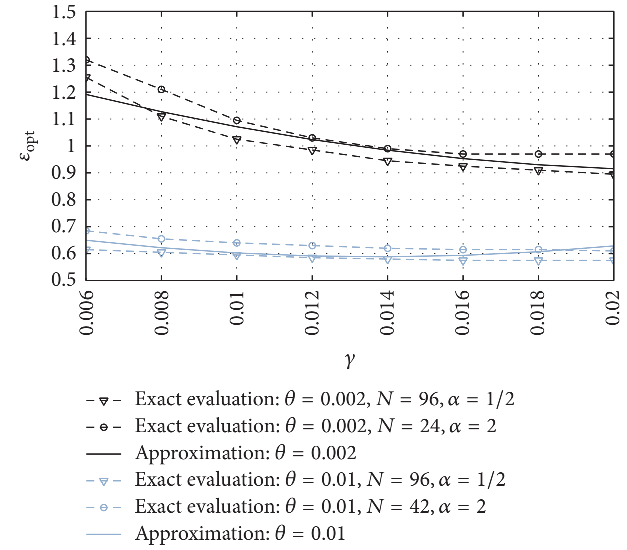


Figure 7: Optimal values of ε: exact versus approximated evaluations ($\lambda = 0.04$, $c = 0.00407$, and $\mu = 0.00255$).

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 Figure 8 (e.g., the video playback starts until window 0 has been completely downloaded). This assumption is valid because some buffering is needed to satisfy QoE targets that are related to interruptions. In Figure 8 we also illustrate the meaning of some of the variables that are defined along this section.

5.1. Probability Distribution of Initial Delay and Mean Downloading Time. We assume here that the distributions of the sojourn time of leeches and the time to download window 0 are both negatively exponential, that is, that their densities are given by $f_{\mathbf{u}}(x) = \theta e^{-\theta x} 1\ (x \geq 0)$ and $f_{\mathbf{v}_0}(x) = c_0 e^{-c_0 x} 1\ (x \geq 0)$, respectively. This is the natural assumption to keep coherence with the constant rates assumed in the fluid model, which in turn are coherent with the idea of considering the system operating in steady state, far from the initial conditions.

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 $\mathbf{v}_0 \leq \mathbf{u}$. Unconditioning, we have that the density of $\mathbf{w}_0$ is

$$f_{\mathbf{w}_0}(x) = (c_0 + \theta) e^{-(c_0+\theta)x} 1 \quad (x \geq 0). \tag{32}$$

Under the assumption that the playback of window 0 will start until it has been successfully downloaded, we can say that (32) also represents the initial delay probability distribution. Furthermore, the mean initial delay is given by

$$T_0 = \frac{1}{c_0 + \theta}. \tag{33}$$

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 sojourn time of a leech.

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 $\mathbf{v}_0$, their common density is given next:

$$f_{\mathbf{r}_0}(x) = \theta e^{-\theta x} 1 \quad (x \geq 0). \tag{34}$$

Although this result could seem a contradiction, it is explained by the fact that now we only focus on those leeches whose sojourn time is large enough to successfully download window 0 and by 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 (32). Furthermore, after applying the previous analysis to every window, 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

$$f_{\mathbf{w}_i}(x) = (c_1 + \theta) e^{-(c_1+\theta)x} 1 \quad (x \geq 0). \tag{35}$$

And the mean time to download window i, given that the leech has not left the system at this point, is simply $T_i = 1/(c_1 + \theta)$. Consequently the mean time to download the whole file, given that the leech did not leave the system is

$$T = \frac{1}{c_0 + \theta} + \frac{N - 1}{c_1 + \theta}. \tag{36}$$

It is interesting to note that if $\alpha = 1$, the required time to download the whole file has an Erlang distribution with rate

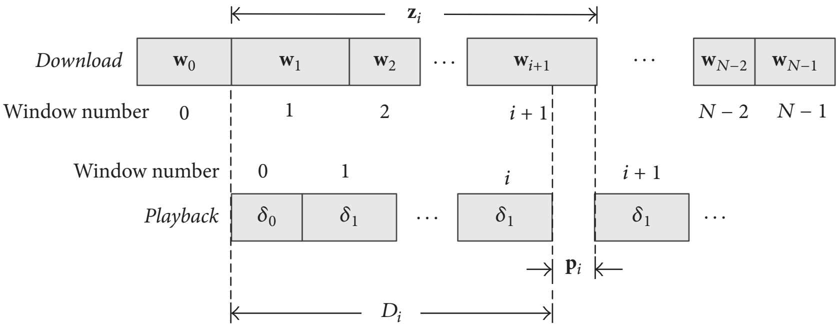


Figure 8: Relationship between download and playback processes.

parameter equal to $c_w + \theta$ and shape parameter equal to N, and (36) is reduced to $T = 1/(c + \theta/N)$. In this expression it is clear that by increasing the number of windows, the mean delay T also increases but only to a maximum value of $1/c$.

5.2. Probability Distribution of Interruption Duration. On the basis of some of the previous results, in this subsection we analyze 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 play back 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 δ_0 the playback time of window 0 and by δ_1 the playback time of any other window (notice that these variables are proportional to d_0 and d_1, resp.). 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 Figure 8). Considering (35), 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 Figure 8, $\mathbf{p}_i$ can be expressed as follows:

$$\mathbf{p}_i = \begin{cases} 0 & \text{if } \mathbf{z}_i \leq D_i, \\ \mathbf{z}_i - D_i & \text{if } \mathbf{z}_i > D_i. \end{cases} \tag{37}$$

Consequently, its probability density function must be given by $f_{\mathbf{p}_i}(x) = F_{\mathbf{z}_i}(D_i)\delta(x) + f_{\mathbf{z}_i}(x + D_i)1$ $(x \geq 0)$, where $F_{\mathbf{z}_i}(x)$ is the cumulative distribution function of $\mathbf{z}_i$ and $\delta(x)$ is Dirac's delta function. Then, the probability density function of the interruption duration after playing window i is given by

$$f_{\mathbf{p}_i}(x) = \left[1 - \sum_{j=0}^{i} \frac{e^{-(c_1+\theta)D_i}\left((c_1+\theta)D_i\right)^j}{j!}\right]\delta(x) + \frac{(c_1+\theta)^{i+1}(x+D_i)^i}{i!} e^{-(c_1+\theta)(x+D_i)} 1 \quad (x \geq 0). \tag{38}$$

Observe 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 given in (38) can be very useful to define QoE targets as functions of the parameters of the system, as we will see in the following section.

6. 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, and delay) are not necessarily correlated to users' experience. Hence, significant effort has been made in order to link them to QoE parameters [22–25]. Particularly, in [22] some experiments were conducted in order to define MOS (one of the most used QoE parameters) as a function of the initial delay and the duration of interruptions that occur along the playback of YouTube videos, one of the most popular services of VoD.

In regard of initial delay (w_0 in our paper), [22] performed different measurements and expressed the MOS as $M_{\text{id}}(w_0) = 5 - 0.862 \log w_0 + 6.718$, where the highest MOS that can be reached is 4.287, occurring when $w_0 = 0$. Since in our model the video playback starts until window 0 has been downloaded, we can evaluate the previous equation by substituting w_0 with $\mathbf{w}_0$. Moreover, if we define an initial delay target MOS, denoted by $M_{\text{id,tgt}}$, we can define a corresponding target initial delay as

$$w_{\text{tgt}} = 10^{(5-M_{\text{id,tgt}})/0.862} - 6.718. \tag{39}$$

Finally, we can evaluate the probability that $M_{\text{id,tgt}}$ is not satisfied, since we know the distribution of $\mathbf{w}_0$:

$$\begin{aligned} q_{\text{id}} &= P\left\{M_{\text{id}}(\mathbf{w}_0) < M_{\text{id,tgt}}\right\} = P\left\{\mathbf{w}_0 > w_{\text{tgt}}\right\} \\ &= e^{-(c_0+\theta)w_{\text{tgt}}}. \end{aligned} \tag{40}$$

In [22] the authors also described the MOS as a function of the interruption duration (p_0), with the following expression: $M_{\text{int}}(p_0) = 1.75e^{-0.334p_0} + 3.19$. In this case, we can also define a target MOS, denoted by $M_{\text{int,tgt}}$. Hence, a target interruption duration can be defined as

$$p_{\text{tgt}} = -\frac{1}{0.334}\ln\left(\frac{M_{\text{int,tgt}} - 3.19}{1.75}\right). \tag{41}$$

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 5 considers interruptions at the end of every window. This means that only one of the random variables $\mathbf{p}_i$ can be used at one time to evaluate a probability analogous to (40). We select $\mathbf{p}_0$ because it has the advantage that it does not depend on other interruptions (since it is the first one) and, most important, because it is the most probable pause, as it can be seen from (38), 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 $M_{\text{int,tgt}}$ is not satisfied, since we know the distribution of $\mathbf{p}_0$; that is,

$$\begin{aligned} q_{\text{id}} &= P\left\{M_{\text{int}}\left(w_0\right) < M_{\text{int,tgt}}\right\} = P\left\{\mathbf{p}_0 > p_{\text{tgt}}\right\} \\ &= e^{-(c_1+\theta)(\delta_0+p_{\text{tgt}})}. \end{aligned} \tag{42}$$

It is relevant to make a comparison between (40) and (42) in terms of the size of window 0. It can be seen from (40) that when c_0 increases (the size of the window decreases), the probability that $M_{\text{id,tgt}}$ is not satisfied is reduced; under the same assumption, the probability of no satisfying $M_{\text{int,tgt}}$ is increased (since δ_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 of elements to properly select c_0 and c_1), it is needed to define target probabilities that $M_{\text{id,tgt}}$ or $M_{\text{int,tgt}}$ are not satisfied, which are denoted by $q_{\text{id,tgt}}$ and $q_{\text{int,tgt}}$, respectively. According to (40) and (42), it can be said that we want that $q_{\text{id}} = e^{-(c_0+\theta)w_{\text{tgt}}} \leq q_{\text{id,tgt}}$ and that $q_{\text{int}} = e^{-(c_1+\theta)(\delta_0+p_{\text{tgt}})} \leq q_{\text{id,tgt}}$ which lead us to

$$d_0 \leq \frac{-w_{\text{tgt}}}{\ln\left(q_{\text{id,tgt}}\right) + \theta w_{\text{tgt}}}, \tag{43}$$

$$d_1 \leq \frac{-\left(\delta_0 + p_{\text{tgt}}\right)}{\ln\left(q_{\text{int,tgt}}\right) + \theta\left(\delta_0 + p_{\text{tgt}}\right)}. \tag{44}$$

As it will be shown in the next subsection, (43) and (44) are used to select the number of chunks in every window in such a way that the QoE parameter set be satisfied.

Another important issue that may be observed while selecting d_0 and d_1 is that while $M_{\text{id}}(w_0)$ is not very susceptible to increases in the initial delay, even small increases in the interruption duration may seriously degrade $M_{\text{int}}(p_0)$ ($M_{\text{id}}(w_0)$ is logarithmic, while $M_{\text{int}}(p_0)$ is exponential). According to [22], this is due to the fact that users are more tolerant to long initial delays than they are to long interruptions.

7. 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 [26, 27]. According to [26], 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.

We now give a brief explanation on how some YouTube service parameters are related to our model. According to [26], 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 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 $r_{\text{cd}} = 2$ for the aforementioned case.

In addition, in [27] it is reported that the average YouTube video duration is 490.5 seconds. Considering this information, we selected as reference for a numerical evaluation a codification rate of 200 kb/s and a chunk size of 64 kBytes, characterizing a video file whose features are illustrated in Figure 9. We denote by M the number of chunks in the file. As such, according to the previous data, for our reference file we chose $M = 192$. 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 [27], 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 θ 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 γ in the range from 0.006 to 0.02; this accounts for seeds remaining in the system an average of 50 to 167 seconds.

Having introduced the video-file parameters, we can establish relations between them and some model variables.

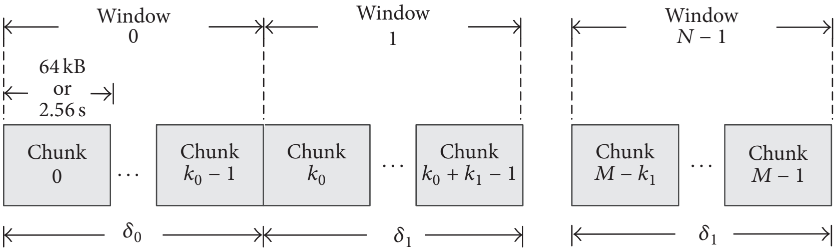


FIGURE 9: Evaluation reference file.

Specifically, $d_0 = k_0/(cM)$, $d_1 = k_1/(cM)$, $\delta_0 = r_{\text{cd}}d_0$, and $\delta_1 = r_{\text{cd}}d_1$.

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

(i) Since we are interested in satisfying a predetermined QoE parameter set, we have to select d_0 and d_1 in such a way that inequalities (43) and (44) are simultaneously satisfied. Since (43) only depends on θ and QoE targets we can select the number of chunks in window 0 by means of the following expression:

$$k_{0,s} = \left\lfloor \frac{-w_{\text{tgt}}cM}{\ln\left(q_{\text{id,tgt}}\right) + \theta w_{\text{tgt}}} \right\rfloor. \quad (45)$$

(ii) Once k_0 is selected, a calculation analogous to (45) can be carried out to select k_1 using (44):

$$k_{1,s} = \left\lfloor \frac{-cM\left(r_{\text{cd}}k_{0,s} + cMp_{\text{tgt}}\right)}{cM\ln\left(q_{\text{int,tgt}}\right) + \theta\left(r_{\text{cd}}k_{0,s} + cMp_{\text{tgt}}\right)} \right\rfloor. \quad (46)$$

(iii) From (45) and (46) we can define the selected values of N and α:

$$N_s = \left\lceil \frac{M - k_{0,s}}{k_{1,s}} \right\rceil + 1,$$
$$\alpha_s = \frac{k_{1,s}}{k_{0,s}}. \quad (47)$$

Notice that $k_{0,s}$ and $k_{1,s}$ are the maximum values that satisfy the QoE parameters set, which means that if lower values were used, N would acquire a larger value than N_s. However, as it is shown in Section 8, the larger the value of N, the larger the value of μ_s required to guarantee abundance.

Assuming that the users' behavior and download/upload parameters are known and using N_s and α_s, it is possible to numerically evaluate (30). This solution provides the optimal values of ε and i. By substituting these values in (27) or (28), depending on the value of i, the minimum value of μ_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 α_s, as well as the remaining parameters, in (24).

8. Numerical Results and Discussions

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 Figure 10 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 Sections 4.1 and 4.2. 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 given in Section 7. 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 (θ takes small values), the value of $\mu_{\min}$ is smaller. Additionally, it is crucial to emphasize that increases in the number of windows significantly increase the required server bandwidth; however, as it is shown later, a very small number of windows have a negative impact on the QoE parameters. In addition to this, Figure 11 shows $\mu_{\min}$ for PWD and two different values of γ. It can be seen that the existence of cooperative seeds (small values of γ) significantly reduces the need of bandwidth provided by servers in the P2P network.

In Figures 12 and 13 we show the number of leeches ($\overline{x}$) and seeds ($\overline{y}$) in equilibrium, respectively. In Figures 14 and 15 we show the system performance in terms of the average initial delay (T_0) and the average video download delay (T), respectively. Those results were obtained through the evaluation of (17), (16), (33), and (36). We also solved an associated Markov chain (see the Appendix) for cross-correlating the results against those in the fluid model.

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 θ entails a large number of leeches, as it is depicted in Figure 12. Since the seeds are peers that finish the file download, the number of them also increases when θ decreases, as it is shown in Figure 13. Indeed, these conditions correspond to a cooperative system which, as previously

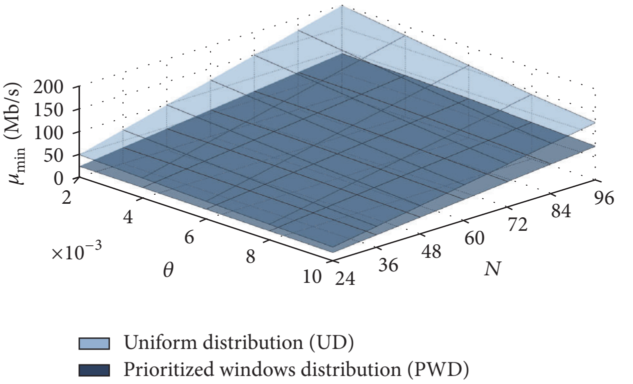


Figure 10: Minimum server bandwidth to achieve abundance, considering $\lambda = 0.04$, $c = 0.00407$, $\mu = 0.00255$, $\gamma = 0.006$, and $\alpha = 1$.

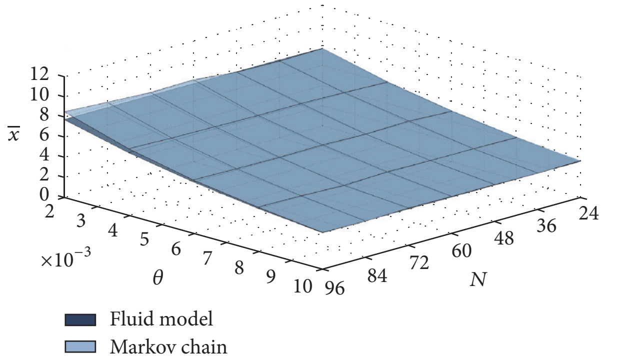


Figure 12: Number of leeches in equilibrium, considering PWD scheme, $\lambda = 0.04$, $\gamma = 0.006$, $c = 0.00407$, $\mu = 0.00255$, $\mu_s = 1.24$, and $\alpha = 1$.

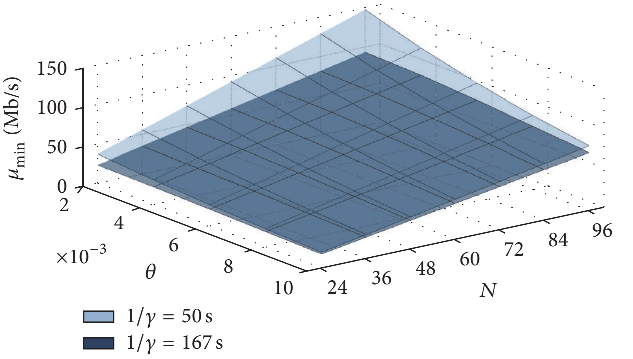


Figure 11: Minimum server bandwidth to achieve abundance, considering the PWD scheme, $\lambda = 0.04$, $c = 0.00407$, $\mu = 0.00255$, and $\alpha = 1$.

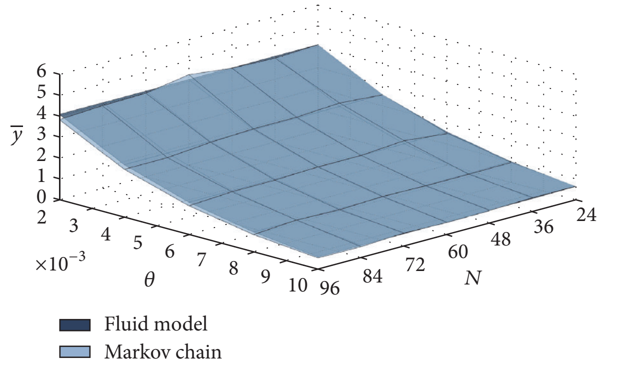


Figure 13: Number of seeds in equilibrium, considering PWD scheme, $\lambda = 0.04$, $\gamma = 0.006$, $c = 0.00407$, $\mu = 0.00255$, $\mu_s = 1.24$, and $\alpha = 1$.

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 Figure 14 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 Figure 10 where a high value of N entails a higher amount of server bandwidth to maintain the system in abundance conditions. As such, in order to reduce the initial delay efficiently, the value of α should be increased instead. Additionally, it must be noticed that θ has no significant effect in T_0, since it is related to both system capacity and resources demand. This last result exhibits the scalability properties of P2P networks.

Regarding the performance of average download delay, T, Figure 15 presents some interesting results that show that under abundance conditions its value is almost a constant. The reason for this is that, in an abundance situation, the required time to download any window is the same (except for window 0 when $\alpha \neq 1$), according to (36). It is important to note that results in Figures 12–15 directly proved that our analytical solution agrees with the numerical one obtained by the Markov chain.

In Figure 16 we evaluate the relation between the QoE parameters that are associated with initial delay and the design parameters α and N. An analogous comparison is shown in Figure 17 for the QoE parameters that are associated with interruptions. As expected, the higher the target MOS ($M_{\text{id,tgt}}$ or $M_{\text{int,tgt}}$), the higher the probability of no satisfaction, q_{id} 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 the effect of α can also be appreciated: when it increases, the size of the initial window decreases and as a consequence q_{id} is improved, but q_{int} is degraded.

In Tables 1 and 2 we summarize the values that were found for the design parameters ε, N, α, and μ_s by following the proposed methodology described at the end of Section 7. Table 1 simply shows that when QoE targets are high, a large amount of server bandwidth must be provided. On the other hand, Table 2 confirms that a P2P network with cooperative seeds (small values of γ) requires a small amount of extra

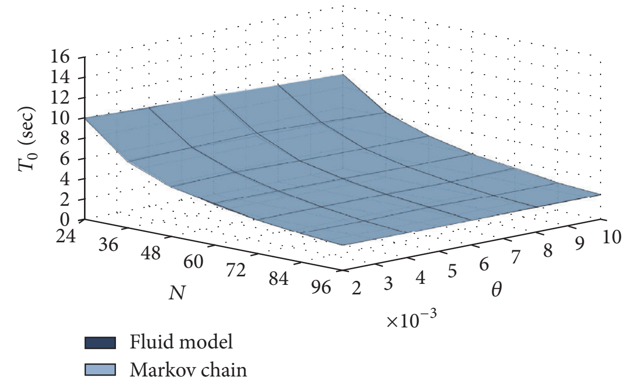


Figure 14: Average time to download window 0 (initial delay), considering PWD scheme, $\lambda = 0.04$, $\gamma = 0.006$, $c = 0.00407$, $\mu = 0.00255$, $\mu_s = 1.24$, and $\alpha = 1$.

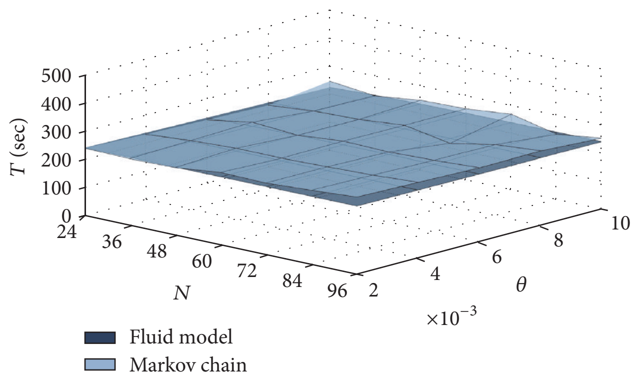


Figure 15: Average time to download the whole file, considering PWD scheme, $\lambda = 0.04$, $\gamma = 0.006$, $c = 0.00407$, $\mu = 0.00255$, $\mu_s = 1.24$, and $\alpha = 1$.

Table 1: Design parameters as functions of the QoE parameters set for $\lambda = 0.04$, $\theta = 0.006$, $\gamma = 0.006$, $c = 0.00407$, and $\mu = 0.00255$.

$M_{\text{id,tgt}}$	$q_{\text{id,tgt}}$	$M_{\text{int,tgt}}$	$q_{\text{int,tgt}}$	ε	N	α	$\mu_{s,pw}$	$\mu_{s,u}$
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

server bandwidth but also shows that when θ 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 noncooperative leeches.

Finally, it must be remarked that we provide the necessary tools and analytical methodology to select the design parameters (ε, N, and α) that satisfy a target set of QoE parameters, given the basic system variables (c, μ, θ, γ, etc.). Furthermore, this is done under a scheme (PWD) that considerably reduces the amount of server bandwidth that is needed to maintain the system in abundance conditions.

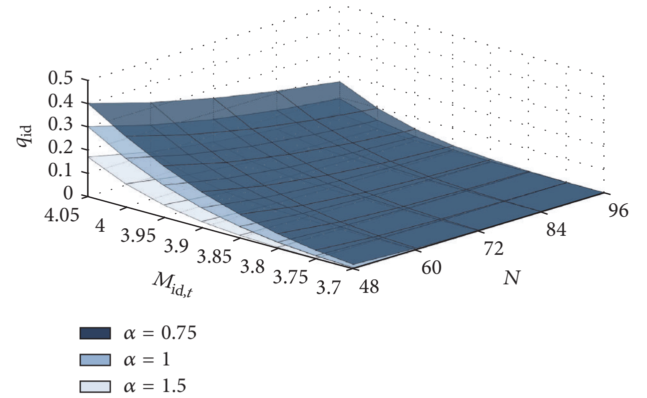


Figure 16: Probability of not satisfying $M_{\text{id},t}$, considering PWD scheme, $\lambda = 0.04$, $\gamma = 0.006$, $c = 0.00407$, $\mu = 0.00255$, and $\mu_s = 1.24$.

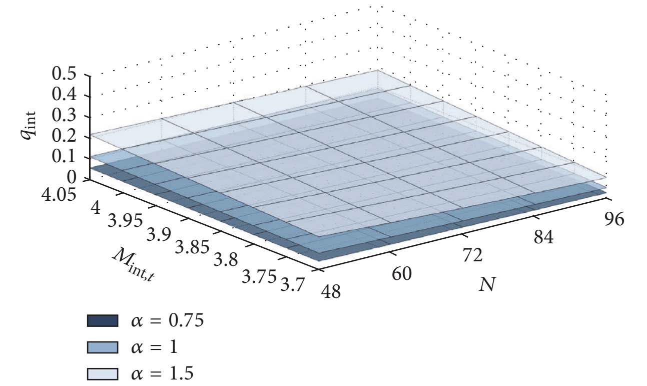


Figure 17: Probability of not satisfying $M_{\text{int},t}$, considering PWD scheme, $\lambda = 0.04$, $\gamma = 0.006$, $c = 0.00407$, $\mu = 0.00255$, and $\mu_s = 1.24$.

9. Conclusions and Future Work

In order to achieve abundance conditions in a P2P network, the existence of cooperative peers is necessary. Since this cooperative scenario does not necessarily exist, it is obviously possible to guarantee an acceptable performance of the system if some servers provide extra upload bandwidth. 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 allows finding a trade-off between the initial delay and the interruption duration,

Table 2: Design parameters in terms of the users' behavior parameters set for $\lambda = 0.04$, $c = 0.00407$, $\mu = 0.00255$, $M_{\text{id,tgt}} = 3.9$, $q_{\text{id,tgt}} = 10\%$, $M_{\text{int,tgt}} = 4.0$, and $q_{\text{int,tgt}} = 10\%$.

θ	γ	ε	N	α	$\mu_{s,pw}$	$\mu_{s,u}$
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

by varying the size of the initial window. This is another contribution of this work, since, to the best of our knowledge, this trade-off has not been previously analyzed in the context of QoE parameters (measured by a MOS metric).

In addition, we also developed an evaluation framework that can be used to calculate the design parameters of the system (number and sizes of windows, upload server bandwidth) that satisfy a set of target QoE-based values, 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 give to their users.

From the results obtained in this work, we have identified a number of issues that have to be addressed in a future work. We are particularly interested in including in our analysis the possibility of pause or moving forward or backward in the video while it is being downloaded and shared, as well as considering varying peer's upload/download bandwidth conditions.

Appendix

Markovian Model

With the fluid model proposed in this paper (Section 3), we can associate a Markovian one as follows. We have now a discrete model of the same system. Consider vector $W(t) = (L_0(t), L_1(t), \ldots, L_{N-1}(t), S(t)) \geq (0, 0, \ldots, 0, 1)$, where $L_i(t)$ and $S(t)$ are, respectively, the number of leeches at window w_i, $i = 0, \ldots, N-1$, and of seeds, at time t. With the standard "exponential assumptions" (plus independence), $W(t)$ is a continuous time homogeneous Markov chain, with initial state $W(0) = (0, 0, \ldots, 0, 1)$. Starting from state $(l_0, l_1, \ldots, l_{N-1}, m)$, for any $i \in \{0, 1, \ldots, N-1\}$, $l_i \geq 0$ and $m \geq 1$, the transition rates are described as follows:

(i) λ, to state $(l_0 + 1, l_1, \ldots, l_{N-1}, m)$,

(ii) $l_i\theta$, to state $(l_0, l_1, \ldots, l_i - 1, \ldots, l_{N-1}, m)$ $(l_i \geq 1)$,

(iii) $\tau^*_{i-1,i}$, to state $(l_0, l_1, \ldots, l_{i-1} - 1, l_i + 1, \ldots, l_{N-1}, m)$ $(1 \leq i \leq N-2$ and $l_{i-1} \geq 1)$,

(iv) $\tau^*_{N-1,N}$, to state $(l_0, l_1, \ldots, l_{N-1} - 1, m + 1)$ $(l_{N-1} \geq 1)$,

(v) $(m-1)\gamma$, to state $(l_0, l_1, \ldots, l_{N-1}, m-1)$ $(m \geq 2)$,

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

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

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

This model can be used to get confidence in the good behavior of the continuous dynamical one, governed by differential equations. The latter has the huge advantage that analytical expressions are often available (the former can only be simulated). The mathematical relations between both approaches are out of the scope of this paper (see [28] for some fundamentals and [29] for the analysis of a P2P case, the one described in [18]).

Most Relevant Variables Summary

c:	File download rate $(c \geq \mu)$
c_0:	Initial window download rate
c_1:	Window i download rate, $i > 0$
$M_{\text{id,tgt}}$:	Target MOS as a function of initial delay
$M_{\text{int,tgt}}$:	Target MOS as a function of interruption duration
N:	Number of windows
q_{id}:	Probability that $M_{\text{id,tgt}}$ is not satisfied
q_{int}:	Probability that $M_{\text{int,tgt}}$ is not satisfied
T:	Required time to download the video file
T_0:	Required time to download the initial window
$\overline{x}$:	Number of leeches in steady state
$\overline{x}_i$:	Number of leeches in window i in steady state
$\overline{y}$:	Number of seeds in steady state
α:	Noninitial window size to initial window size ratio
γ:	Departure rate for a seed
ε:	Priority control parameter
θ:	Departure rate for a leech
λ:	Arrival rate of peers
μ:	File upload rate
μ_0:	Initial window upload rate
μ_1:	Window i upload rate, $i > 0$
μ_{min}:	Minimum servers upload bandwidth that guarantees abundance in the whole system
μ^i_{min}:	Minimum servers upload bandwidth that guarantees abundance in window i
μ_s:	Servers upload bandwidth
τ_i:	Transition rate from window i to window $i+1$
τ_{N-1}:	Transition rate from window $N-1$.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Y.-W. Huang and Y.-C. Chen, "A study on content delivery strategy for QoE enhancement in P2P IPTV," in *Proceedings of the 14th IEEE International Conference on Computer and Information Technology (CIT '14)*, pp. 520–527, Xi'an, China, September 2014.

[2] D. Kondo, Y. Hirota, A. Fujimoto, H. Tode, and K. Murakami, "P2P live streaming system for multi-view video with fast switching," in *Proceedings of the 16th International Telecommunications Network Strategy and Planning Symposium, Networks 2014*, 7, 1 pages, September 2014.

[3] O. Ojo, A. Oluwatope, and O. Ogunsola, "UStream: ultra-metric spanning overlay topology for peer-to-peer streaming systems," in *Proceedings of the 17th IEEE International Symposium on Multimedia (ISM '15)*, pp. 601–604, Miami, Fla, USA, December 2015.

[4] Y. Zhou, T. Z. J. Fu, and D. M. Chiu, "A unifying model and analysis of P2P VoD replication and scheduling," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1163–1175, 2015.

[5] P. Romero, F. Robledo, P. Rodríguez-Bocca, and C. Rostagnol, "Analysis and design of peer-assisted video-on-demand services," *International Transactions in Operational Research*, vol. 21, no. 4, pp. 559–579, 2014.

[6] W. Wu, R. T. B. Ma, and 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, pp. 612–621, 2014.

[7] M. Faiqurahman and A. I. Kistijantoro, "Implementation of modified Probabilistic Caching Schema on Bittorrent protocol for video on demand content," in *Proceedings of the 16th International Seminar on Intelligent Technology and Its Applications (ISITIA '15)*, pp. 357–362, IEEE, Surabaya, Indonesia, May 2015.

[8] M. Wichtlhuber, S. Dargutev, S. Müller, A. Klein, and D. Hausheer, "QTrade: a quality of experience based peercasting trading scheme," in *Proceedings of the IEEE International Conference on Peer-to-Peer Computing (P2P '15)*, Boston, Mass, USA, September 2015.

[9] T. Rohmer, A. Nakib, and 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, pp. 350–362, 2014.

[10] R. Dubin, O. Hadar, Y. Freifeld et al., "Hybrid clustered peer-assisted DASH-SVC system," in *Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM '15)*, pp. 1651–1656, October 2015.

[11] D. Ciullo, V. Martina, M. Garetto, E. Leonardi, and G. L. Torrisi, "Peer-assisted VoD systems: an efficient modeling framework," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1852–1863, 2014.

[12] "Global Internet Phenomena. Latin America and North America Report 2015," Internet: https://www.sandvine.com.

[13] Ericsson Mobility Report, June 2016, https://www.ericsson.com/mobility-report.

[14] S. Jia, Y. Ma, Y. Zhang, T. Wang, and M. Zhang, "Modelling of P2P-based video sharing performance for content-oriented community-based VoD systems in wireless mobile networks," *Mobile Information Systems*, vol. 2016, Article ID 1319497, 13 pages, 2016.

[15] C. Xu, S. Jia, L. Zhong, and G.-M. Muntean, "Socially aware mobile peer-to-peer communications for community multimedia streaming services," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 150–156, 2015.

[16] E. Esquivel, M. E. Rivero-Angeles, and G. Rubino, "Priority scheme for window-based video-on-demand transmission on BitTorrent-like Peer-to-Peer networks," in *Proceedings of the IEEE International Conference on Communications (ICC '13)*, pp. 3000–3005, Budapest, Hungary, June 2013.

[17] M. E. Rivero-Angeles, G. Rubino, I. O. O. Torres, and L. A. Martinez, "Window-based streaming video-on-demand transmission on BitTorrent-like peer-to-peer networks," in *Proceedings of the IEEE 10th Consumer Communications and Networking Conference (CCNC '13)*, pp. 1–6, IEEE, January 2013.

[18] M. E. Rivero-Angeles and G. Rubino, "Priority-based scheme for file distribution in peer-to-peer networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–6, May 2010.

[19] S. Tewari and L. Kleinrock, "Analytical model for BitTorrent-based live video streaming," in *Proceedings of the 4th Annual IEEE Consumer Communications and Networking Conference (CCNC '07)*, pp. 976–980, IEEE, Las Vegas, Nev, USA, January 2007.

[20] C. Liang, Z. Fu, Y. Liu, and C. W. Wu, "Incentivized peer-assisted streaming for on-demand services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1354–1367, 2010.

[21] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '04)*, Portland, Ore, USA, October 2004.

[22] T. Hossfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial delay vs. interruptions: between the devil and the deep blue sea," in *Proceedings of the 4th International Workshop on Quality of Multimedia Experience (QoMEX '12)*, pp. 1–6, IEEE, Yarra Valley, Australia, July 2012.

[23] P. T. A. Quang, K. Piamrat, and C. Viho, "QoE-aware routing for video streaming over ad-hoc networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '14)*, pp. 181–186, December 2014.

[24] S. Mohamed and G. Rubino, "A study of real-time packet video quality using random neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1071–1083, 2002.

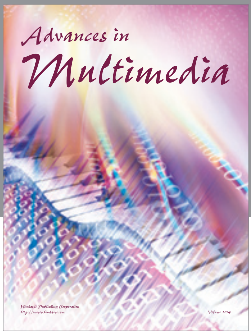
[25] K. D. Singh, Y. Hadjadj-Aoul, and G. Rubino, "Quality of experience estimation for adaptive HTTP/TCP video streaming using H.264/AVC," in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC '12)*, pp. 127–131, Las Vegas, Nev, USA, January 2012.

[26] J. Ramos-Munoz, J. Prados-Garzon, P. Ameigeiras, J. Navarro-Ortiz, and J. Lopez-Soler, "Characteristics of mobile youtube traffic," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 18–25, 2014.

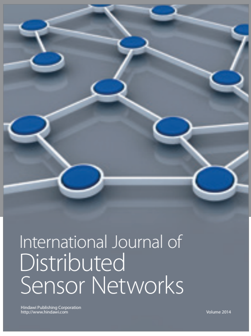
[27] G. Dimopoulos, P. Barlet-Ros, and J. Sanjuàs-Cuxart, "Analysis of YouTube user experience from passive measurements," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM '13)*, pp. 260–267, Zurich, Switzerland, October 2013.

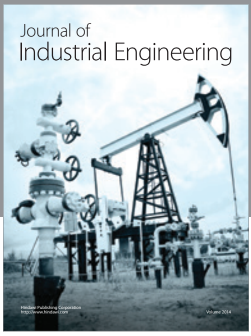
[28] S. N. Ethier and T. Kurtz, *Markov Processes: Characterization and Convergence*, John Wiley & Sons, New York, NY, USA, 1986.

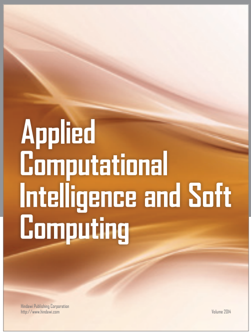
[29] L. Aspirot, E. Mordecki, and G. Rubino, "Fluid limits applied to peer to peer network analysis," in *Proceedings of the 8th International Conference on Quantitative Evaluation of Systems (QEST '11)*, pp. 13–20, IEEE, Aachen, Germany, September 2011.

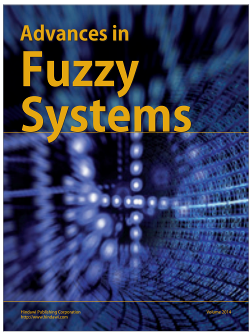

Advances in
Multimedia

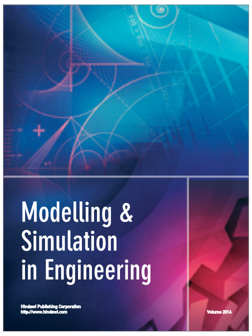
The Scientific
World Journal

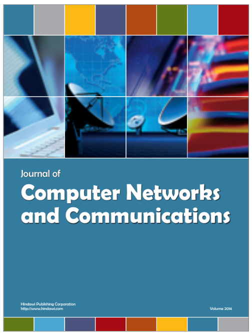

International Journal of
Distributed
Sensor Networks

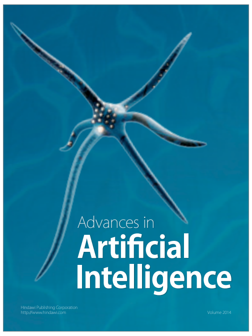

Journal of
Industrial Engineering

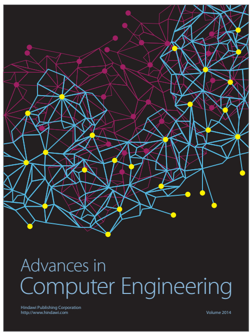

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

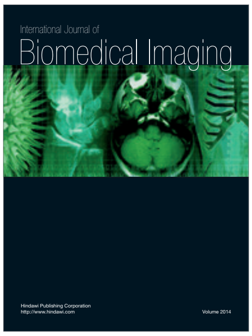

Modelling &
Simulation
in Engineering

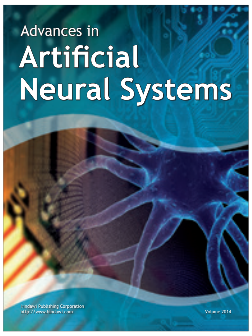

Journal of
Computer Networks
and Communications

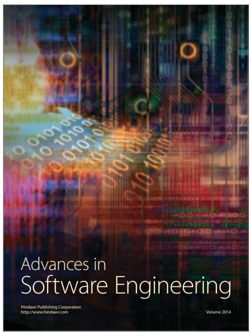

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

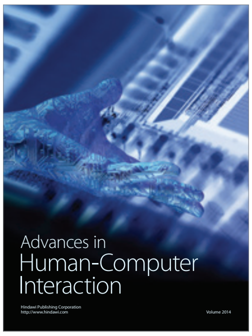
International Journal of
Computer Games
Technology

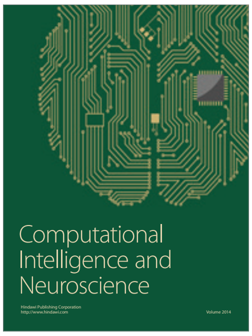

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering