Scalable playback rate control in P2P live streaming systems

Maria Efthymiopoulou*, Nikolaos Efthymiopoulos*, Athanasios Christakidis*, Nikolaos Athanasopoulos#,Spyros Denazis*, Odysseas Koufopavlou*

*Department of Electrical and Computer Engineering

University of Patras

Patras, Greece

Department of Electrical Engineering

Eindhoven University of Technology,

Eindhoven, The Netherlands

Email: {mefthymiop, nefthymiop, schristakidis}@ece.upatras.gr, n.athanasopoulos@tue.nl, {sdena, odysseas} @ece.upatras.gr

***Abstract*—Current commercial live video streaming systems are based either on a typical client-server (cloud) or on a peer-to-peer (P2P) architecture. The former architecture is preferred for stability and QoS, provided that the system is not stretched beyond its bandwidth capacity, while the latter is scalable with small bandwidth and management cost. In this paper, we propose a P2P live streaming architecture in which by adapting dynamically the playback rate we guarantee that peers receive the stream even in cases where the total upload bandwidth changes very abruptly. In order to achieve this we develop a scalable mechanism that by probing only a small subset of peers monitors dynamically the total available bandwidth resources and a playback rate control mechanism that dynamically adapts playback rate with respect to the aforementioned resources. We model analytically the relationship between the playback rate and the available bandwidth resources by using difference equations and in this way we are able to apply a control theoretical approach. We also quantify monitoring inaccuracies and dynamic bandwidth changes and we calculate analytically the amount of bandwidth resources that we keep idle, by lowering the playback rate, in order to mitigate their negative effects. Finally, we evaluate the control strategy and the theoretical model in a packet level simulator of a complete P2P live streaming system that we designed in OPNET Modeler [11]. Our evaluation results show the uninterrupted and complete stream delivery (every peer receives more than 99% of video blocks in every scenario) even in very adverse bandwidth changes.**

Keywords—peer to peer (P2P), live streaming, scalable monitoring, robust control

I. INTRODUCTION

Video streaming systems are producing a considerable percentage of internet traffic. According to Cisco [7] will exceed 90% of the globally consumed Internet traffic by 2015. On the other hand, major streaming service providers (e.g. YouTube) suffer from high bandwidth costs. Peer-to-peer live streaming and video on demand architectures [6], [16],[17],[18],[19] have received a lot of research attention in the past few years aiming at achieving a better trade-off between bandwidth costs and quality of the transmitted video, while providing scalability and stability of these services. In more detail, the major requirements for P2P live streaming systems are:

Efficiency of the media distribution in terms of utilization of peers' upload bandwidth, in order to minimize any additional bandwidth contributed by a set of media servers (cloud) and/or maximize the playback rate of the video which they are able to deliver. Efficiency has a direct impact on the trade-off between bandwidth costs and the video quality that the system is able to distribute.

Stability of the system which is defined as the uninterrupted and complete stream delivery in each participating peer in the presence of dynamic conditions (unrelated network traffic, bandwidth changes, peer arrivals and/or departures), which affect the amount of the available upload bandwidth of the participating peers. These dynamic conditions may have serious impact on the quality of experience (QoE) perceived by the users.

Scalability which is determined by the amount of resources (bandwidth, storage, processing overhead) that cloud, which manage the system, has to contribute in order to sustain the uninterrupted delivery of the stream, as the number of participating peers grows. The design of a scalable system requires low amount of resources even if the number of participating peers is high.

In order to increase the efficiency of P2P live streaming systems research community focused on the development of P2P overlay architectures. P2P overlay is represented by a graph in which each node represents a user, and each edge that connects two nodes represents the exchange of media blocks between them. In P2P live streaming systems each peer is connected, through the underlying network, with a small subset of peers. Several P2P overlay architectures [1], [2], [3], [9], [14] have been proposed to optimize the P2P overlay. The main idea is to exploit efficiently the heterogeneous upload bandwidth of each participating peer. Despite their efficiency when the total upload bandwidth of participating peers happens to exceed the bandwidth requirements of live streaming, these architectures are static. Consequently they have two major drawbacks in the presence of dynamic conditions. The first is the interruption of successful stream delivery under lack of upload bandwidth. The second drawback concerns the non-utilization of idle upload bandwidth, in case that there is surplus of upload bandwidth, for the distribution of higher video quality.

There are two strategies in order to adapt the peer to peer live streaming to the dynamic upload bandwidth conditions of participating peers. The first is to dynamically allocate upload bandwidth from auxiliary sources (e.g. clouds) and the second is to adapt the playback rate according to the existing upload bandwidth of participating peers. The selection of a strategy has to do with the QoE that participating peers desire and the business model of the service provider. In case that users and the service provider desire a costless live steaming the second strategy has to be selected. In case that they desire a live streaming with high QoE the first strategy has to be selected. In this work we cope up only with the second strategy (adaptation of the playback rate). We believe that the proposed modeling and control can be extended to be used also in the first strategy or a hybrid one but we leave this as our future work.

Towards the first strategy, the research community has proposed monitoring systems such as [5], [8] that use stochastic methods for dynamically monitor total available bandwidth resources in a P2P overlay and calculate the deficit or surplus of the total upload bandwidth in it, in order to allocate bandwidth. These systems are scalable but suffer from three drawbacks. The first is that the stochastic methods are suitable only for specific bandwidth distributions among participating peers. The second is that, due to their low confidence interval, their efficiency in terms of peer bandwidth exploitation is often low. The third and the most important is that they are not stable in cases of sudden disturbances (i.e. underlying network traffic changes and/or peers' arrivals-departures).

Motivated by the lack of a critical mass of research in the area of the second strategy of dynamic resource monitoring and dynamic playback rate control for P2P live streaming systems and by the serious issues raised above, we establish a new idle bandwidth monitoring and playback rate control system that:

- Provides a scalable monitoring, in terms of bandwidth and processing overhead that it costs to the live streaming service provider, by exploiting balancing properties of P2P overlays and distributed block transmission schedulers for P2P live streaming.
- It ensures uninterrupted stream distribution even in a highly dynamic environment as it is based on an analytical model that we designed which factorizes the disturbances which a dynamic environment introduces to a P2P live streaming system and exploits modern control theory to mitigate their effects.
- It utilizes the upload bandwidth of participating peers. In specific, we calculate analytically the maximum playback rate that the system is able to deliver as a function of the accuracy of the measurements and the maximum possible disturbance in total available upload bandwidth.

To the best of our knowledge this is the first and only study towards these objectives.

The reminder of this paper is structured as follows: Section II presents our P2P live streaming system's architecture. Section III provides the problem setting. Section IV presents the nominal model and control strategy. Section V enriches the model of the system by factorizing also measurement inaccuracies, model inaccuracies and dynamic network conditions and performs robust stability analysis. In Section VI we calculate the reference point as a function of the aforementioned factors and we theoretically prove the capability of our system to guarantee the delivery of the stream with the maximum possible quality. Section VII evaluates the proposed system, while Section VIII analyzes related work. Finally in Section IX we conclude and highlight our future steps.

II. ARCHITECTURE

Our P2P live video streaming system consists of a media server in a cloud, (noted by S) and a set of peers (noted by N). The video stream that the system disseminates is divided into video blocks. The cloud is responsible for: i) the initial diffusion of video blocks to a small subset of nodes among participating peers, ii) the tracking of the network addresses of a small set of participating peers in order to assist the construction and management of the P2P overlay, iii) the dynamic and scalable monitor of the resources of participating peers, iv) the dynamic control of the playback rate.

In order to allow peers to exchange video blocks, each peer maintains network connections with a small subset of other peers which will be noted as neighbors. The sets of these connections change dynamically and form a dynamic graph called the P2P overlay. In our previous works [2], [3] we present a graph topology and P2P overlay management (dynamic and distributed optimization) algorithms that each peer periodically executes which result in the dynamic reconfiguration of the P2P overlay. We use distributed optimization theory in order to dynamically ensure in a distributed (scalable) and dynamic fashion that: i) peers have connections proportional with their upload bandwidth, ii) peers have connections with other peers close to the underlying network, iii) our P2P overlay is adaptable to underlying network changes and peer arrivals and departures. This allows us to efficiently exploit all the available bandwidth resources even if they are highly heterogeneous.

The dynamic output of the P2P overlay management algorithms that run in each participating peer is a neighbor list that is passed in the Distributed Block Transmission Scheduler.

After that, video block exchanges are coordinated by the Distributed Block Transmission Scheduler (DBTS) which is comprised by a set of algorithms executed by every peer who dynamically communicates with its neighbors. The major objective of DBTS is to ensure the timely delivery of every block to every peer by exploiting the upload bandwidth of participating peers and the additional bandwidth resources that media servers may contribute. Each peer periodically sends to its neighbors control messages which encapsulate information about video blocks that it owns. Thus, periodically each peer (through a matching algorithm) is able to request from each one of its neighbors a different video block or nothing if there is no video block to request. In order to perform the requests a matching algorithm is executed periodically by each peer and its objective is to request as many unique blocks as possible. These requests are served sequentially by each peer who prioritizes them by selecting each time its most deprived neighbor to serve its block request. As most deprived is defined the neighbor that has the smallest total number of blocks compared to video blocks that sender peer owns. Our proposed DBTS is analyzed in detail in our previous works [2], [3]. DBTS sends the video blocks that have to be sent in the P2P congestion control component and the ordered stream with the blocks that it receives to the video player.

Our proposed P2P overlay and our DBTS enhance our P2P live streaming system with two properties that we exploit to model our system. The first property (**Property 1**) is that if idle bandwidth exists it is derived from bandwidth surplus in the system and not from the inefficiency of the system to exploit it. In other words we guarantee that the presence of idle bandwidth implies the complete stream delivery. The second property is that the percentages of the idle resources among participating peers are almost equal (**Property 2**).

Our P2P congestion control mechanism is able to manage sequential transmissions of video blocks to multiple locations that DBTS sends to it and to provide to the Scalable Bandwidth Monitoring the dynamic estimation of the upload bandwidth capacity and the idle bandwidth resources of each participating peer with the way that will be requested from the latter.

The Scalable Bandwidth Monitoring is a centralized component in the cloud that applies a monitoring protocol which aggregates the monitoring information from DBTS and P2P congestion control and forms all the required metrics that Playback Rate Control needs.

In the rest of this paper by exploiting the features of the aforementioned components we analyze a dynamic and scalable playback rate control architecture that calculates dynamically the playback rate towards the control of the idle bandwidth resources. Then, video blocks are generating dynamically with the desired playback rate as this is determined by the playback rate control strategy and passing to DBTS in order to be diffused to all participating peers. Figure 1 illustrates our proposed P2P live streaming system architecture.

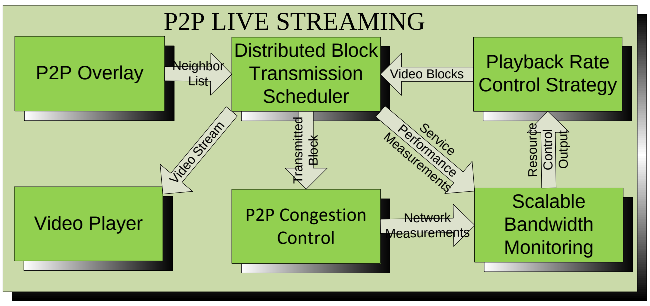


Fig.1. Proposed P2P live streaming system architecture

III. PROBLEM STATEMENT

We assume a set of peers *N* that receive a stream of the same media object (video). In order to receive the stream all peers in *N* issue requests to their neighbors (a small subset of N) with a bit rate p_k which is the media object playback rate. The subscript k is an integer denoting the time instant. The fulfillment of these requests represents the incoming flows in the system. These requests are served from the same set of peers which exploit their upload bandwidth. These are the outgoing flows in the system. By exploiting them P2P congestion control is able to calculate dynamically: i) upload bandwidth, $u_{(i)k}$, of each participating peer , ii) an idle percentage of the upload bandwidth capacity, $id_{(i)k}$, of each participating peer i in N.

The first problem that we solve in this work is the development of an analytical model that connects the playback rate with the idle percentage of the upload bandwidth of participating peers. The second problem that we solve is to exploit the properties of our system in order to create a scalable monitoring system that will allow us to estimate dynamically and with accuracy the idle resources of the whole system by probing a number of M peers much lower than N and stable as N grows. The third problem is to create a control strategy with which we exploit our analytical model in order control $id_{(i)k}$ of each participating peer in N to a reference value id_{REF} by adapting dynamically p_k. This allows our system to ensure the on time distribution of every video block to every participating peer. The fourth problem that we solve is to calculate analytically the minimum id_{REF} that guarantees the stable distribution of the stream by taking into account the inaccuracies that our model introduces and the disturbances of the system (dynamic changes in $u_{(i)k}$ of peers in N). In this way the system operates in the maximum playback rate whose complete distribution is able to guarantee. The final question that we answer is to find the upper bound of $id_{(i)k}$ in order to predict in advance the maximum percentage of bandwidth resources that possibly remain idle quantified again as a function of the inaccuracies that our model introduces and the disturbances of the system.

IV. PLAYBACK RATE CONTROL – NOMINAL CASE

Playback rate control (PRC) is a control functionality. RPC is executed periodically, at a time instant k, with a period T. It is executed in a centralized fashion by the server, S, who generates the media object that is streamed. Its objective is to set the idle time percentage $id_{(i)k}$ of each peer i in N to a reference value id_{REF}, by periodically adjusting p_k. In the rest of this section we model this process analytically and we propose a control strategy with which we periodically calculate p_k. In order to achieve this we use a set of symbols. These symbols are presented altogether in Table I below. As the reader may have observed the index i in brackets indicates a peer i that belongs to N and the index k represents a time instant k.

In this section in order to describe progressively the model of our system we make two assumptions that we remove in section V. These are:

- **Assumption 1**: According to **Property 2** that we describe in Section II $id_{(i)k}=id_k$, for each i belongs to N. We note that id_k represents the average $id_{(i)k}$ in N. Under this assumption we can estimate id_k if we measure $id_{(i)k}$ of only one random peer in N.

- **Assumption 2**: Period T, with which PRC is executed, is lower than the time interval that is needed for significant changes in the total upload bandwidth of participating peers. So we can do the approximation that total upload bandwidth remains the same between two consecutive executions of PRC.

At any time instant k and in case that there are sufficient upload bandwidth resources the P2P overlay and the DBTS (Section II) guarantee the complete delivery of the stream to every peer in the set N and that the incoming flow to each participating peer is p_k. Consequently the sum of the incoming flows of N peers is Np_k. Without loss of generality the sum of the incoming flows that peers receive is equal to the sum of the outgoing flows that peers in N contribute by using their upload bandwidth. The sum of the outgoing flows is the sum of their non-idle upload capacity $u_{(i)k}$. By quantifying the above and by exploiting **Property 1** of section II we derive (1) as shown below:

TABLE I . NOTATION

Symbol	Definition
S	Generator (source) of the media object
N	Set of participating peers (in the equations below we use N as the number of participating peers)
p_k	Media playback rate at time instant k
$u_{(i)k}$	Upload capacity (upper limit) of peer i at time instant k
$id_{(i)k}$	Idle time percentage of peer i that at time instant k between 0 and 1
id_k	Average estimated idle time percentage of N peers at time instant k between 0 and 1
id_{REF}	Average idle time percentage reference value that is between 0 and 1
T	Period of execution of PRC
w_k	System input that repsesents the change in the playback rate that is determined from PRC
w_{REF}	System input in the equilibrium point
δ1	Percentage of modeling and monitoring inaccuracies. It belongs to ($-\delta 1_M, \delta 1_M$)
δ2	Percentage of average upload bandwidth change. It belongs to ($-\delta 2_M, \delta 2_M$)

$$Np_k = \sum_{i \in N} (1 - id_{(i)k}) u_{(i)k} (1)$$

Under **Assumption 1** we can rewrite (1) as:

$$Np_k = (1 - id_\kappa) \sum_{i \in N} u_{(i)k} (2)$$

Rewriting (2) for time instant k+1, we have:

$$Np_{k+1} = (1 - id_{k+1}) \sum_{i \in N} u_{(i)k+1} (3)$$

Now by dividing (2), (3) under **Assumption 2** we have:

$$(1-id_{k+1})p_k = p_{k+1}(1-id_k)(4)$$

By definition at time instant k+1, the dynamic playback rate, p_k, is expressed as the sum of the playback rate at time instant p_k and w_k. Thus, holds that:

$$p_{k+1} = p_k + w_k(5)$$

By combining (4), (5) we have:

$$id_{k+1}p_k = (p_k + w_k)id_k - w_k(6)$$

Setting $id_k=id_{k+1}=id_{REF}$ in (6) we obtain w_{REF} which can be defined as the input in the equilibrium point and is equal to 0. So the equilibrium point is (id_{REF} ,0). In order to have a system which has as its equilibrium point (0,0). We now set:

$$x_k = id_k - id_{REF}(7)$$

The idle time percentage id_k belongs to the interval (0,1) by definition. Thus x_k ranges between ($-id_{REF}$, $1-id_{REF}$). By substituting (7) for k=k and for k=k+1 in (6) we have:

$$x_{k+1} = x_k + (\frac{x_k + (id_{REF} - 1)}{p_k})w_k(8)$$

We observe that (8) is nonlinear. In order to have a linear closed loop system we select a feedback linearization [15] control strategy. Feedback linearization is a strategy that introduces a state feedback such that the closed loop system becomes linear. To this end we select a control strategy $w_k(x_k,p_k)$ of the form:

$$w_k = \frac{p_k}{x_k + (id_{REF} - 1)}(k_c - 1)x_k(9)$$

In (9) k_c is a parameter that we choose. By combining now (8) and (9) we have the linear system with eigenvalue k_c which is:

$$x_{k+1} = k_c x_k(10)$$

In this way is easy to see from (10) that the series $\{x_k\}$ converges to 0, and so id_k to id_{REF}, for any value of k_C that belongs to (-1, 1).

Since k_c is a designer's choice we can explicitly set the eigenvalue of the system by just setting k_c. The implementation of the proposed control strategy is:

$$w_k = \frac{p_k}{id_k - 1}(k_c - 1)(id_k - id_{REF})(11)$$

V. ROBUST PLAYBACK RATE CONTROL (RPRC)

In this section we remove the two assumptions of the previous sections. Assumption 1 is removed because P2P Overlays and DBTSs as i.e. [1],[14],[2],[3] try to balance in a distributed and dynamic fashion the block requests to the participating peers achieving in this way similar but not identical idle percentages ($id_{(i)k}$ of each peer i in N). Under this observation we remove **Assumption 1** and in this way we rewrite precisely (2) in (12).

$$N_\kappa p_k = (1 - id_k + \delta 1)\sum_{i \in N} u_{(i)k}(12)$$

In order to calculate id_k we use (13) and we derive it from the average $id_{(i)k}$ by sampling M out N peers (with M<<N). So δ1 expresses the inaccuracy that our modeling introduces and δ1 belongs to the interval ($-\delta 1_M$,$\delta 1_M$) where $\delta 1_M$ is a small positive number.

$$id_k = \sum_{i \in M} id_{(i)k} / |M| (13)$$

$$(1 - \sum_{i \in M} id_{(i)k} / M + \delta 1)\sum_{i \in N} u_{(i)k} = \sum_{i \in N}(1 - id_{(i)k})u_{(i)k}(14)$$

If we could measure every time instant $id_{(i)k}$ and $u_{(i)k}$ of each peer i that belongs to N we could calculate δ1 each time instant k by solving (14). So δ1 depends on the dynamic variance that P2P Overlay and DBTS introduces in various $id_{(i)k}$. The estimation of δ1 has to do with the level of heterogeneity among participating peers and in the size of the sets M and N. In Section VII by using the P2P overlay and the DBTS from [2],[3] we run various simulations even under extreme levels of peer's upload bandwidth heterogeneity and for various values of M and N. In this way we are able to calculate offline (experimentally) $\delta 1_M$ which is the maximum absolute of δ1 ever observed for a specific M under various upload bandwidth distributions and sizes of the set N.

Next we remove **Assumption 2** in order to be precise we define as δ2 possible increase or decrease in the percentage of the average upload bandwidth within T and so we rewrite precisely the ratio between the average upload bandwidth at time instant k and the average upload bandwidth at time instant k+1 as:

$$[\sum_{i \in N_k} u_{(i)k} / \mathrm{N}_\kappa] / [\sum_{i \in N_{k+1}} u_{(i)k+1} / \mathrm{N}_{\kappa+1}] = 1 + \delta 2(15)$$

Again $\delta 2$ belongs to $[-\delta 2_M, \delta 2_M]$ where $\delta 2_M$ is again a small positive number. The variable $\delta 2$ represents the maximum possible change of the percentage of the average upload bandwidth during a period T and depends on the underlying network conditions and the user behavior. Its calculation is out of the scope of this paper and it's a factor that expresses the tradeoff between stability and efficiency in a real system. On the other hand the proposed control strategy does not make any assumption on the value of $\delta 2_M$ and so it can be considered transparent to it and suitable for any value of $\delta 2_M$.

Although we broke the two assumptions that we did in the previous section we can still use (3) as it expresses precisely our objective to find the playback rate p_{k+1} that will set idle bandwidth percentage of every participating peer to id_{REF} at time instant k+1. By dividing now (3) with (12) and by the use of (15) we have:

$$p_k / p_{k+1} = [(1 - id_k + \delta 1) / (1 - id_{k+1})](1 + \delta 2) (16)$$

By rearranging (16) with respect to id_{k+1} and by the use of (5) we obtain:

$$id_{k+1} = (id_k - 1 - \delta 1)(1 + w_k / p_\kappa)(1 + \delta 2) + 1 (17)$$

From (17) by using again (7) and (8) we have:

$$x_{k+1} = (x_k + id_{REF} - 1 - \delta 1)(1 + w_k / p_\kappa)(1 + \delta 2) + 1 - id_{REF} (18)$$

And by applying the same control strategy from (9) we have:

$$x_{k+1} = [(1 + \delta 2)(x_k + id_{REF} - 1 - \delta 1)(k_c x_k + id_{REF} - 1) / (x_k + id_{REF} - 1)] + 1 - id_{REF} (19)$$

VI. REFERENCE POINT CALCULATION

If $\delta 1_M$ and $\delta 2_M$ where equal to zero the system would be the nominal system. In that case we could set id_{REF} to a very small positive value (i.e 2%) and guarantee (according to **Property 1** of our system) in this way the complete delivery of a stream with playback rate almost equal (i.e. 98%) with the total upload bandwidth. Our modeling and measurement techniques introduce an inaccuracy because they designed also for scalability. This inaccuracy is factorized with $\delta 1$. Additionally the real world conditions in the underlying network lead to dynamic changes of the average upload bandwidth (15) and we factorize them with $\delta 2$.

From (19) we observe that the uncertainties $\delta 1$, $\delta 2$ not allow to set x_k and so id_k to a specific reference point. To this end in this section we exploit (19) and the bounds $\delta 1_M$, $\delta 2_M$ of $\delta 1$, $\delta 2$ respectively to calculate an interval inside which x_k and so id_k reside for given $\delta 1_M$, $\delta 2_M$ and id_{REF}. In this way we can find for which values of id_{REF} for which id_k remains always positive and in this way according to **Property 1** we are able to guarantee the complete delivery of the stream. Finally we calculate the minimum id_{REF} for which id_k remains always positive and in this way we are able to drive our system to a state that it delivers the maximum possible playback rate that is able to guarantee.

The rest of this section is structured as follows. In Lemma 1 we calculate x_M which is the maximum x_k that the system reaches as a function of $\delta 1_M$,$\delta 2_M$ and id_{REF}. In Lemma 2 we calculate x_E which is the minimum x_k that the system reaches as a function of $\delta 1_M$,$\delta 2_M$ and id_{REF}. In Lemma 3 we exploit Lemma 1 and Lemma 2 and we calculate the minimum id_{REF} which guarantees that id_k will always remain positive and the interval in which id_k will be always moving.

Lemma 1: Let us assume x_M such that:

$$x_M = [(1+\delta 2_M)(1-id_{REF}) - \sqrt{(1+\delta 2_M)^2 (id_{REF}-1)^2 + 4\delta 1_M (1+\delta 2_M)(id_{REF}-1)}\,]/2 \quad (20)$$

If $x_k<x_M$ then $x_{k+1}<x_M$ under the constraints that $x_M>\delta 2_M(1-id_{REF})$, $x_M<1-id_{REF}-\delta 1_M$ and $id_{REF}\leq 1-4\delta 1_M(1+\delta 2_M)$

Proof:

Let's assume that:

$$x_{k+1} < x_M \quad (21)$$

By the use of (19) we have:

$$[(1+\delta 2)(x_k + id_{REF} - 1 - \delta 1)(id_{REF}-1) + (1-id_{REF}-x_M)(x_k+id_{REF}-1)]/(x_k+id_{REF}-1) < 0 \quad (22)$$

Let us set:

$$f(x_k,\delta 1,\delta 2) = [(1+\delta 2)(x_k+id_{REF}-1-\delta 1)(id_{REF}-1) \\ +(1-id_{REF}-x_M)(x_k+id_{REF}-1)]/(x_k+id_{REF}-1) \quad (23)$$

If we now calculate the derivative of $f(x_k,\delta 1,\delta 2)$ as to $\delta 1$ we have (24). If we now consider that $\delta 2>-1$, id_{REF} belongs to (0,1) by definition and x_k belongs to ($-id_{REF}$, $1-id_{REF}$) then the aforementioned derivative is negative for each $\delta 1$ that belongs to ($-\delta 1_M$, $\delta 1_M$). This is a helpful observation because if $f(x_k,\delta 1,\delta 2)$ is negative for $\delta 1=-\delta 1_M$ it is negative for each $\delta 1$ that belongs to ($-\delta 1_M$, $\delta 1_M$).

$$\partial f(x_k,\delta 1,\delta 2)/\partial \delta 1 = -(1+\delta 2)(id_{REF}-1)/(x_k+id_{REF}-1) \quad (24)$$

Again if we calculate the derivative of $f(x_k,\delta 1,\delta 2)$ as to $\delta 2$ we have (25). This derivative is negative for each $\delta 2$ that belongs to ($-\delta 2_M$, $\delta 2_M$) if additionally $x-id_{REF}-1-\delta 1$ is negative and so if and only if $x_M+id_{REF}-1+\delta 1_M<0$ (which requires $x_M<1-id_{REF}-\delta 1_M$). So we now know that if $f(x_k,\delta 1,\delta 2)$ is negative for $\delta 2=-\delta 2_M$ it is negative for each $\delta 2$ that belongs to ($-\delta 2_M$, $\delta 2_M$).

$$\partial f(x_k,\delta 1,\delta 2)/\partial \delta 2 = (x_k+id_{REF}-1-\delta 1)(id_{REF}-1)/(x_k+id_{REF}-1) \quad (25)$$

After the observations above for δ1 and δ2 concerning (24) and (25) we can say now that (22) is true if and only if (26) is true.

$$[(1-\delta 2_M)(x_k + id_{REF} - 1 + \delta 1_M)(id_{REF} - 1) + (1 - id_{REF} - x_M)(x_k + id_{REF} - 1)] / (x_k + id_{REF} - 1) < 0 (26)$$

As the denominator of (26) is negative for each x_k belongs to ($-id_{REF}$, $1-id_{REF}$) the nominator of (26) has to be positive for each $x_k<x_M$. We rewrite now in (27) the nominator of (26) as a linear function of x_k and so (22) is true if and only if (27) is true.

$$[-\delta 2_M (id_{REF} - 1) - x_M] x_k + [(id_{REF} - 1)(id_{REF} + \delta 1_M) + 1 - id_{REF} - x_M] > 0 (27)$$

As a liner function of x_k if (27) has the multiplier of x_k negative as depicted in (28) and is positive for for $x_k=x_M$ as depicted in (29) then (27) is true for each $x_k<x_M$ and so the same holds for (22).

$$-\delta 2_M (id_{REF} - 1) - x_M < 0 (28)$$

$$(1-\delta 2_M)(x_M + id_{REF} - 1 + \delta 1_M)(id_{REF} - 1) + (1 - id_{REF} - x_M)(x_M + id_{REF} - 1) > 0 (29)$$

We can easily observe that (29) is a trinomial as to x_M and the minimum x_M for which (29) is true is presented in (30) under the constraint that depicted in (31), which ensures that the quantity under the root in (30) is positive.

$$x_M = [(1+\delta 2_M)(1 - id_{REF}) - \sqrt{(1+\delta 2_M)^2 (id_{REF} - 1)^2 + 4\delta 1_{\mathrm{M}} (1+\delta 2_M)(id_{REF} - 1)}\,] / 2 (30)$$

As we already analyzed in order to have a positive quantity under the root in (30) we derive after calculations (31)

$$id_{REF} \le 1 - 4\delta 1_M / (1+\delta 2_M) (31)$$

Until now we proved that if we select x_M as presented in (30) and the constraints in (28) and (31) hold then (27) and so (26) are true for any $x_k<x_M$. If now $x_M<1-id_{REF}-\delta 1_M$, in order to have negative derivative of $f(x_k,\delta 1,\delta 2)$ as to δ2, then (22) is also true and so (21). In this way Lemma 1 has been proved.

Lemma 2: Let's assume x_E such that:

$$x_{\mathrm{E}} = (1+\delta 2_{\mathrm{M}})(x_{\mathrm{M}} + id_{REF} - 1 - \delta 1_{\mathrm{M}})(id_{REF} - 1) / (x_{\mathrm{M}} + id_{REF} - 1) + 1 - id_{REF} (32)$$

If $x_k<x_M$, then $x_{k+1}>x_E$ under the constraints that $x_E<-\delta 2_M(1-id_{REF})$ and $x_M<1-id_{REF}-\delta 1_M$

Proof: Let's assume that:

$$x_{k+1} > x_E \quad (33)$$

By using (19) we must prove that:

$$[(1+\delta 2)(x_k + id_{REF} - 1 - \delta 1)(id_{REF} - 1) + (1 - id_{REF} - x_E)(x_k + id_{REF} - 1)] / (x_k + id_{REF} - 1) > 0 \quad (34)$$

Let us set:

$$\begin{aligned} g(x_k, \delta 1, \delta 2) &= [(1+\delta 2)(x_k + id_{REF} - 1 - \delta 1)(id_{REF} - 1) \\ &+ (1 - id_{REF} - x_E)(x_k + id_{REF} - 1)] / (x_k + id_{REF} - 1) \quad (35) \end{aligned}$$

If we now calculate the derivative of $g(x_k,\delta 1,\delta 2)$ as to $\delta 1$ we have (36). If we now consider that $\delta 2>-1$, id_{REF} belongs to (0,1) and by definition x_k belongs to ($-id_{REF}$, $1-id_{REF}$) then the aforementioned derivative is negative for each $\delta 1$ that belongs to ($-\delta 1_M$, $\delta 1_M$). This is a very helpful observation because we know now that if $g(x_k,\delta 1,\delta 2)$ is positive for $\delta 1=\delta 1_M$ it is positive for each $\delta 1$ that belongs to ($-\delta 1_M$, $\delta 1_M$).

$$\partial g(x_k, \delta 1, \delta 2) / \partial \delta 1 = -(1+\delta 2)(id_{REF} - 1) / (x_k + id_{REF} - 1) \quad (36)$$

Again if we calculate the derivative of $g(x_k,\delta 1,\delta 2)$ as to $\delta 2$ we have (37). This derivative is negative for each $\delta 2$ that belongs to ($-\delta 2_M$, $\delta 2_M$) if additionally $x-id_{REF}-1-\delta 1$ is negative and so if and only if $x_M+id_{REF}-1+\delta 1_M<0$ (which requires $x_M<1-id_{REF}-\delta 1_M$).So now know that if $g(x_k,\delta 1,\delta 2)$ is positive for $\delta 2=\delta 2_M$ it is positive for each $\delta 2$ that belongs to ($-\delta 2_M$, $\delta 2_M$).

$$\partial g(x_k, \delta 1, \delta 2) / \partial \delta 2 = (x_k + id_{REF} - 1 - \delta 1)(id_{REF} - 1) / (x_k + id_{REF} - 1) \quad (37)$$

After these observations we conclude that (34) is true if and only if (38) is true.

$$[(1+\delta 2_M)(x_k + id_{REF} - 1 - \delta 1_M)(id_{REF} - 1) + (1 - id_{REF} - x_E)(x_k + id_{REF} - 1)] / (x_k + id_{REF} - 1) > 0 \quad (38)$$

As the denominator of (38) is negative for each x_k belongs to ($-id_{REF}$, $1-id_{REF}$) the nominator of (38) has to be negative for each $x_k<x_M$. We rewrite now the nominator of (38) as a linear function of x_k in (39).

$$[-\delta 2_M(1 - id_{REF}) - x_E]x_k + [(1+\delta 2_M)(id_{REF} - 1 - \delta 1_M)(id_{REF} - 1) + (1 - id_{REF} - x_E)(id_{REF} - 1)] < 0 \quad (39)$$

If the multiplier of x_k in (39) is positive as we depict in (40) and if the nominator of (38) is negative for $x_k=x_M$ as we depict in (41) then (38) is positive for each $x_k<x_M$.

$$-\delta 2_M(1-id_{REF})-x_E>0(40)$$

$$(1+\delta 2_M)(x_M+id_{REF}-1-\delta 1_M)(id_{REF}-1)+(1-id_{REF}-x_E)(x_M+id_{REF}-1)>0(41)$$

But after calculations the maximum x_E for which (41) is true is depicted in (42).

$$x_E=(1+\delta 2_M)(x_M+id_{REF}-1-\delta 1_M)(id_{REF}-1)/(x_M+id_{REF}-1)+1-id_{REF}(42)$$

Until now we proved that if we select x_E as presented in (42) and the constraint in (40) holds then (39) and so (38) are true for any $x_k<x_M$. If now $x_M<1-id_{REF}-\delta 1_M$, in order to have negative derivative of $g(x_k,\delta 1,\delta 2)$ as to $\delta 2$, then (34) is also true and so (33). In this way Lemma 2 has been proved.

Lemma 3: If $id_{REF}=-x_E$ and $x_E<x_0<x_M$ then always $0<id_k<x_M+id_{REF}$
Under the constraints that: i)$x_M<1-id_{REF}-\delta 1_M$, ii)$x_E<-\delta 2_M(1-id_{REF})$, iii)$x_M>\delta 2_M(1-id_{REF})$,iv) $id_{REF}<1-4\delta 1_M(1+\delta 2_M)$
Proof:
If $x_k<x_M$ and the four aforementioned constraints hold then from Lemma 1 and Lemma 2 $x_E<x_{k+1}<x_M$. Since $x_0<x_M$ by induction we have that always $x_E<x_k<x_M$. Where x_0 the measurement of x_k at time instant 0. By using now (7) we have (43).

$$x_E<id_k-id_{REF}<x_M(43)$$

If we now put in (43) $x_E=-id_{REF}$ then we derive (44).

$$0<id_k<x_M+id_{REF}(44)$$

Under the observation of (44) Lemma 3 has been proved.

The corollary of Lemma 3 is that for given $\delta 1_M$, $\delta 2_M$ and by setting $id_{REF}=-x_E$ we are able to calculate the minimum reference point (id_{REF}) which guarantees that id_k will remain positive and so we guarantee (according to **Property 1** of our system) for these $\delta 1_M$, $\delta 2_M$ complete distribution of the stream with the maximum possible playback rate. Additionally we are able to calculate the interval in which id_k is moving in this case.

In Table II we present the minimum id_{REF} which we can select according to Lemma 3 as a function of $\delta 1_M$ and $\delta 2_M$. According to Lemma 3 by setting $id_{REF}=-x_E$ id_k will be in always in the interval $(0,x_M+id_{REF})$. In Table III we present the upper threshold of id_k again as a function of $\delta 1_M$ and $\delta 2_M$.

TABLE II. MINIMUM ID_{REF} AS A FUNCTION OF THE PERCENTAGE OF MODELING AND MONITORING INACCURACIES AND PERCENTAGE OF AVERAGE UPLOAD BANDWIDTH CHANGE

$\delta 2_M \backslash \delta 1_M$	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07
0	0	0.011	0.021	0.031	0.042	0.054	0.065	0.078
0.01	0.011	0.021	0.031	0.042	0.053	0.064	0.076	0.088
0.02	0.021	0.031	0.041	0.052	0.063	0.075	0.086	0.099
0.03	0.031	0.041	0.052	0.063	0.074	0.085	0.097	0.11
0.04	0.04	0.05	0.061	0.072	0.083	0.095	0.107	0.12
0.05	0.049	0.059	0.07	0.081	0.093	0.105	0.117	0.13
0.06	0.058	0.068	0.079	0.091	0.102	0.114	0.127	0.14
0.07	0.067	0.077	0.089	0.1	0.112	0.124	0.137	0.15
0.08	0.076	0.087	0.098	0.109	0.121	0.134	0.147	0.16
0.09	0.084	0.095	0.106	0.118	0.13	0.143	0.156	0.169
0.1	0.092	0.103	0.114	0.126	0.139	0.151	0.165	0.179
0.11	0.101	0.112	0.124	0.136	0.148	0.161	0.175	0.189
0.12	0.109	0.12	0.132	0.144	0.157	0.17	0.184	0.198
0.13	0.117	0.128	0.14	0.153	0.165	0.179	0.193	0.207
0.14	0.124	0.135	0.148	0.16	0.173	0.187	0.201	0.216
0.15	0.132	0.144	0.156	0.169	0.182	0.196	0.21	0.225

TABLE III. UPPER BOUND OF ID_K AS A FUNCTION OF THE PERCENTAGE OF MODELING AND MONITORING INACCURACIES AND PERCENTAGE OF AVERAGE UPLOAD BANDWIDTH CHANGE

$\delta 2_M \backslash \delta 1_M$	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07
0	0.001	0.022	0.042	0.062	0.084	0.106	0.13	0.154
0.01	0.021	0.041	0.062	0.083	0.105	0.127	0.15	0.174
0.02	0.041	0.061	0.081	0.102	0.124	0.147	0.169	0.194

0.03	0.06	0.08	0.101	0.122	0.143	0.165	0.189	0.213
0.04	0.078	0.098	0.119	0.14	0.162	0.184	0.208	0.232
0.05	0.097	0.116	0.137	0.158	0.181	0.203	0.226	0.251
0.06	0.115	0.134	0.155	0.177	0.198	0.221	0.245	0.269
0.07	0.132	0.152	0.173	0.194	0.216	0.239	0.263	0.287
0.08	0.15	0.17	0.19	0.211	0.233	0.257	0.28	0.305
0.09	0.167	0.187	0.207	0.229	0.251	0.274	0.298	0.322
0.1	0.183	0.203	0.224	0.245	0.268	0.29	0.315	0.34
0.11	0.2	0.22	0.241	0.262	0.284	0.307	0.331	0.357
0.12	0.216	0.236	0.257	0.278	0.3	0.323	0.348	0.373
0.13	0.232	0.251	0.272	0.294	0.316	0.34	0.364	0.389
0.14	0.247	0.267	0.288	0.309	0.332	0.355	0.38	0.406
0.15	0.262	0.283	0.303	0.325	0.347	0.371	0.395	0.421

VII. EVALUATION

In order to evaluate our system we conducted packet level simulations in OPNET Modeler [11]. The video is divided in 10 video blocks per second and each one of them disseminated from the media server S to 5 random peers. In order to diffuse each video block to every participating peer on time we use the DBTS and the P2P overlay as described in Section II.

As mentioned in Section V, $\delta 1$ expresses the inaccuracy of our modeling and depends on the variance of $id_{(i)k}$ that the P2P Overlay and DBTS introduce and on the level of heterogeneity of upload bandwidth of participating peers. In order to derive $\delta 1_M$ exploited our P2P live streaming model that constructed in OPNET Modeler in which (in contrast with a real system) we are able to measure every time instant k the values of $u_{(i)k}$ and $id_{(i)k}$. By solving (14) as to $\delta 1$ we have:

$$\delta 1 = [\sum_{i \in N}(1 - id_{(i)k}) * u_{(i)k}] / \sum_{i \in N} u_{(i)k} - 1 + \sum_{i \in M} id_{(i)k} \,/\, M \quad (45)$$

We exploit our simulation environment and by using the values of $u_{(i)k}$ and $id_{(i)k}$ in (45) we are able to calculate $\delta 1$ every time instant k. Initially in Fig. 2 we demonstrate the Cumulative Distribution Function (CDF) of the values of $\delta 1$ in various time instants during the distribution of a video with duration 200 sec and stable playback rate p=900kbps. In this scenario N=300, M=60 and we have a stable and homogeneous environment (participating peers have all stable and the same upload bandwidth which is equal with 1000kbps). From Fig.2 we can observe that $\delta 1$ is always in the interval (-0.02, 0.02) which means that in this scenario the inaccuracy that our modeling introduces is always less than 2% and so $\delta 1_M$=0.02.

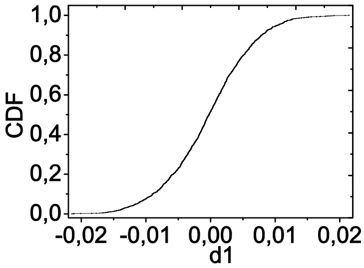


Fig.2 CDF of δ1 (Homogenous Environment, N=300,M=60)

By using this methodology we made the estimation of $\delta1_M$ through extensive simulations under various values of N and M and in various environments.

With the term environment is characterized the upload bandwidth distribution of participating peers. In the rest of this section will used two upload bandwidth distributions of participating peers. The first is noted homogenous environment and in this case all the peers have equal upload bandwidth which may change over time or not. The second is noted as heterogeneous environment and in this case apeers upload bandwidth differs and still may change over time or not. In order to evaluate our system in a highly heterogeneous environment, to prove the uninterrupted stream delivery even in this case, did not selected the peer's upload bandwidths to follow a normal or uniform distribution but a distribution with much higher statistical variance. More specifically in our heterogeneous environment 50% of peers have a minimum quantity of upload bandwidth u_{MIN}, 25% of peers have upload bandwidth $2u_{MIN}$ and 25% of peers have upload bandwidth $4u_{MIN}$. For instance initially when u_{MIN}=500Kbps, 50% of participating peers have upload bandwidth 500 Kbps, 25% of participating peers have 1000 Kbps and 25% 2000 Kbps. Even a uniform distribution in which upload bandwidths would be from 500 Kbps to 2000 Kbps would have led to much lower variance.

In all scenarios u_{MIN} changes over time in order to achieve dynamic average upload bandwidth but the analogy of upload bandwidth between the three aforementioned categories of peers remains. In the rest of this work we chose to present the homogenous environment as it offers an environment almost ideal for our proposed algorithms and we chose this specific heterogeneous environment as it offers a very adverse upload bandwidth distribution and consequently an exhaustive test for our proposed algorithms.

In Fig. 3 and Fig. 4 we depict $\delta1_M$ in four different scenarios where 100,200,300 and 400 peers respectively join the system the same time and started watching a video of 200 seconds with constant playback rate of p=900 Kbps and constant average upload bandwidth equal with 1000kbps. Fig. 3 corresponds to homogeneous environment and Fig.4 corresponds to heterogeneous environment. The size of the sample M in each scenario varies to 20, 40, 60 and 80 peers. $\delta1_M$ is the maximum absolute of δ1 ever observed for a specific set of M and N. According to Fig.3 and Fig.4, $\delta1_M$ decreases while the size of the sample M increases. Moreover, for the same size of M and various number of N $\delta1_M$ remains almost constant. This makes us optimistic on the accuracy of our system as the number N of participating peers grows. From Fig.3 we observe that in homogenous environment a selection of M=40 can give us accuracy around 2% even when we have 400 participating peers. From Fig. 4 we observe that even in a highly heterogeneous environment a selection of M=40 can give us accuracy around 6% even when we have 400 participating peers.

From Fig. 3 and Fig. 4 we can derive the scalability properties of our system. As stated before, from both figures observed that as the number of participating peers (N) grows the monotony in not increasing and so the inaccuracy with which we estimate the idle resources of our system does not increases and remains stable. For example in Fig. 3 when we probe 40 peers (M=40) the inaccuracy that is introduced is always between 2,5% and 3% as the number of participating grows from 100 to 400 with a non-increasing tendency.

In the rest of this section we will use M=50 and we will consider that $\delta 1_M$=0.06 in order to cover the worst case scenario. By considering that our proposed control algorithm is executed with a period T= 7 seconds the monitoring of our control algorithm will be around 7 UDP packets per second and irrelevant with the number of participating peers.

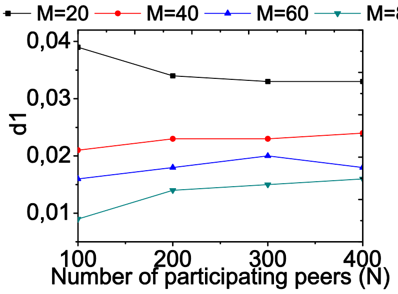


Fig.3 $\delta 1_M$ over N and M in homogeneous environment

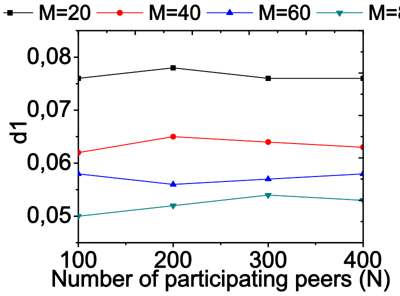


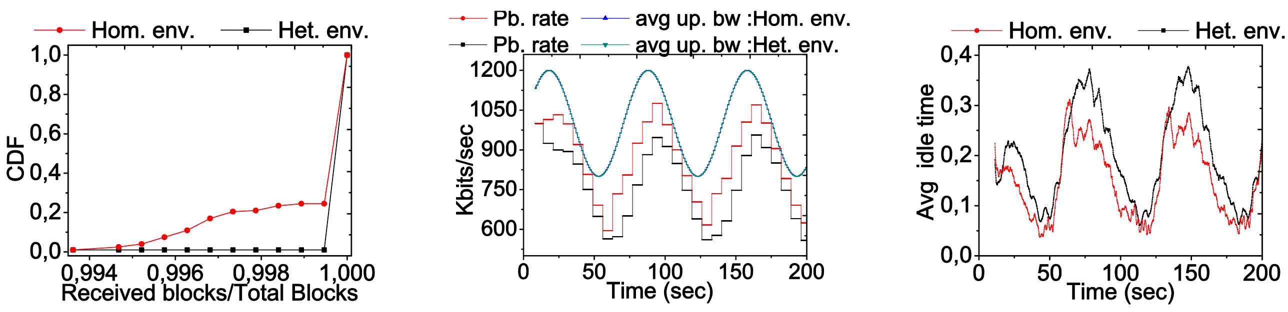
Fig.4 $\delta 1_M$ over N and M in heterogeneous environment

In the rest of this section in order to evaluate the performance of our system we present five different scenarios where the average upload bandwidth of the participating peers changes over time (with various patterns) and the playback rate changes dynamically time according to (11). Playback rate control is executed with a period T=7sec and number of participating peers N is 400. For the first three scenarios, $u_{i(k)}$ of every peer i in N changes according to a sinusoidal function, while the video duration is again 200 seconds. For the rest two scenarios, $u_{i(k)}$ of every peer i in N changes according to a linear function and the video duration is 100 seconds. In these five scenarios we present the evaluation of our system for both homogeneous and heterogeneous environment.

As it can be seen in the rest of this section all the selected scenarios are very adverse and $\delta 2_M$ sometimes reaches 0.14. After these observations we can select from Table II id_{REF} which will be equal with 0.201 and predict that id_k will be moving in the interval (0,0.38).

In each scenario three types of graphs are used in order to present our results. The first type (5a-9a) is the Cumulative Distribution Function (CDF) of the percentage of video blocks that each peer receives during the whole experiment. In the second type (5b-9b) we depict the way in which the average upload bandwidth changes over time and the way in which our PRC adapts the playback rate (p_k) to the bandwidth changes. Both are measured in kbps and so we can present them in the same vertical axis. Finally, the third type (5c-9c) represents average idle upload bandwidth percentage (id_k) over time. In each one of the graphs we present the behavior of our PRC in the homogenous and the heterogeneous environment for the same average upload bandwidth change pattern.

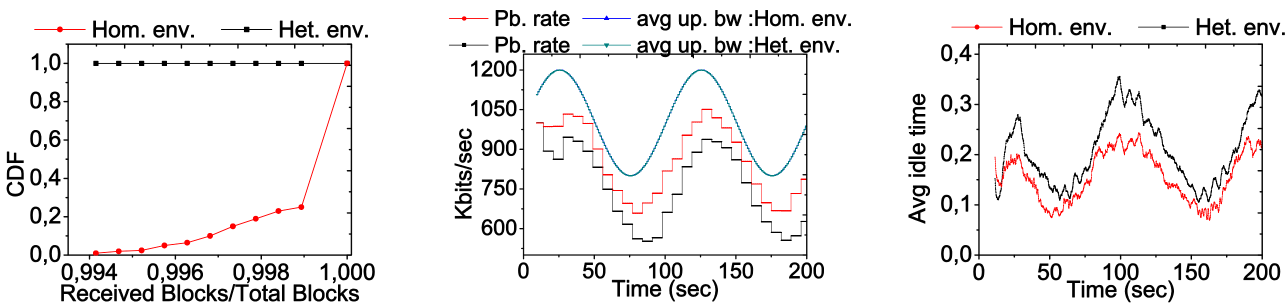
In Scenario I, each peer's upload bandwidth $u_{i(k)}$ changes according to sinusoidal function with width 200kbps and period 70 seconds. Fig.5 summarizes our results. From Fig. 5a we observe that all peers receive more than 99% of the stream with a mean around 99,9%. This proves the capability of our PRC to guarantee the successful delivery of the stream in this adverse bandwidth changing pattern. From Fig. 5b we observe that even during these heavy transactional periods the playback rate never exceeds the average upload bandwidth and is never lower than around 30% of the average upload bandwidth. In the heterogeneous environment the playback rate is 0-10% lower than that in the homogenous and this due to the lower accuracy of our model in highly heterogeneous cases. We consider that this is a minor drawback of our system as it happens only temporarily in cases of high upload bandwidth disturbances. From Fig. 5c we observe that id_k is always positive, as our theoretical model predicts, and that reveals why all peers receive almost completely the video stream. Additionally id_k is always lower than 38% which is the upper limit that our theoretical model predicts.



(a) (b) (c)

Fig.5 Scenario I: a)CDF of the ratio of the received video blocks among participating peers, b) average upload bandwidth and playback rate over time, c)average idle over time

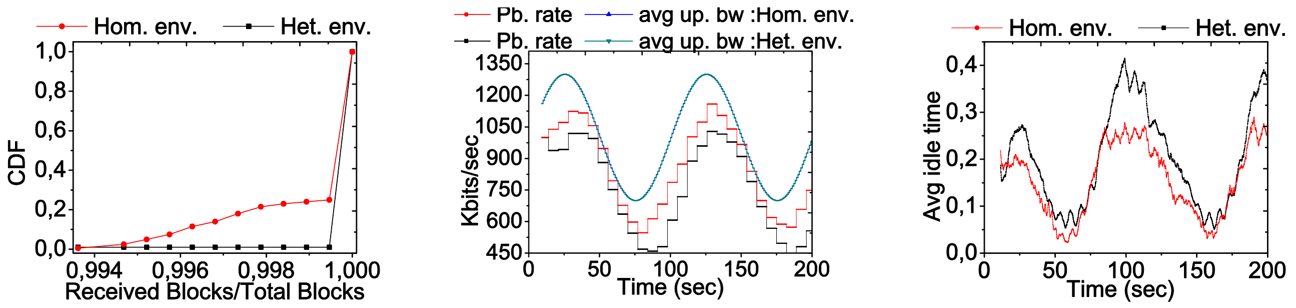
In Scenario II, each peer's upload bandwidth $u_{i(k)}$ changes according to sinusoidal function with width 200kbps and period 100 seconds. Fig.6 summarizes our results. Again we observe in Fig. 6a the reception of the stream from all peers (the mean is 99,9%). The bandwidth change pattern is smoother than the previous scenario and so smoother is also the playback rate changes in Fig. 6b with better (smoother) performance in homogenous and in heterogeneous case. Again from Fig. 6c we observe that id_k remains always higher than 0 and lower than 35%.



(a) (b) (c)

Fig.6 Scenario II: a)CDF of the ratio of the received video blocks among participating peers, b) average upload bandwidth and playback rate over time, c)average idle over time

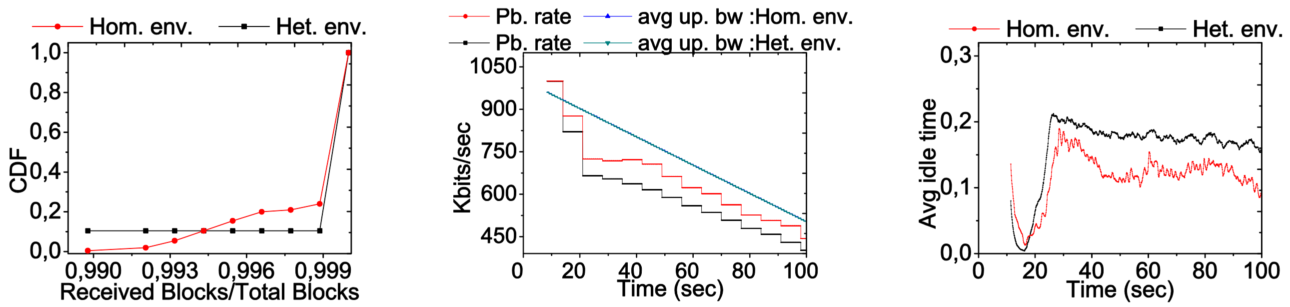
In Scenario III, each peer's upload bandwidth $u_{i(k)}$ changes according to sinusoidal function with width 300kbps and period 100 seconds. Fig.7 summarizes our results. Here we have more adverse bandwidth change than the previous two scenarios but the reception of the stream remains complete (in Fig 7a we observe that all peers receive more than 99,4% with a mean higher than 99,9%) and id_k always positive (Fig 7c). This is due to the fact that the playback rate manages to follow the bandwidth changes pattern (Fig 7b)



(a) (b) (c)

Fig.7 Scenario III: a)CDF of the ratio of the received video blocks among participating peers, b) average upload bandwidth and playback rate over time, c)average idle over time

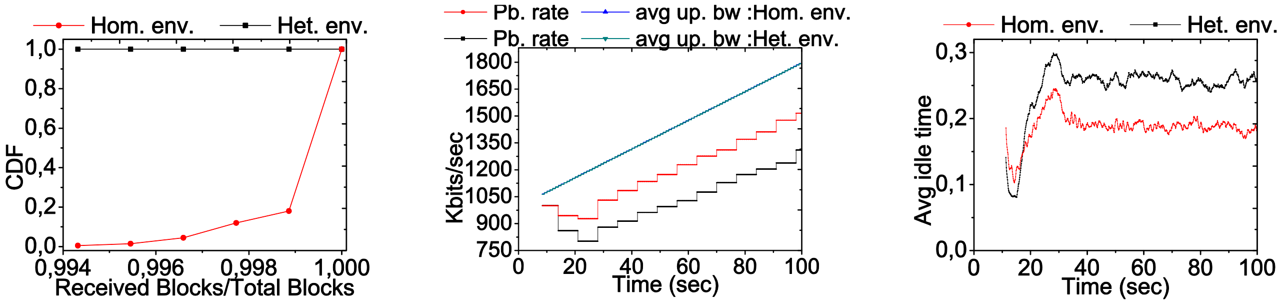
In Scenario IV, we evaluate the case of a decrease of upload bandwidth. So the average upload bandwidth of participating peers for both homogeneous and heterogeneous environment decreases linearly from 1000 Kbps (similar to the playback rate) to 500 Kbps during 100 seconds. Fig. 8 summarizes our results. The complete reception of the stream that is depicted in Fig. 8a and the ability of the playback rate to follow the bandwidth change pattern (Fig. 8b) testify the ability of our PRC to follow the linear and continuous bandwidth decrease.



(a) (b) (c)

Fig.8 Scenario IV: a)CDF of the ratio of the received video blocks among participating peers, b) average upload bandwidth and playback rate over time, c)average idle over time

In Scenario V, we evaluate the case of an increase of upload bandwidth. So the average upload bandwidth of participating peers for both homogeneous and heterogeneous environment increases linearly from 1000 Kbps (similar to the playback rate) to 1800 Kbps during 100 seconds. Fig.9 summarizes our results. This is the easiest scenario for the distribution of the stream (Fig. 9a) for our PRC as the bandwidth only increases. We observe that our PRC follows the increase (Fig 9b) and avoids having much idle bandwidth (Fig. 9c).



(a) (b) (c)

Fig.9 Scenario V: a)CDF of the ratio of the received video blocks among participating peers, b) average upload bandwidth and playback rate over time, c)average idle over time

Overall, from Fig. 5a-9a arises that every peer acquires more than 99% (with a mean more than 99,9%) of video blocks which means that our system ensures the stable delivery of the stream even during adverse bandwidth changes. Fig. 5b-9b indicate how the playback rate p_k is adjusted dynamically over time according to the changes of the average upload bandwidth of participating peers. It is emphasized here, that, all selected scenarios are very adverse and our controller manages to follow these changes. From Fig. 5c-9c could be observed average peers' idle time over time. The assumption is that the average idle time (id_k) remains in the predicted from our theoretical model interval.

Finally, it is important to highlight the negative impact that has a highly heterogeneous environment (increased $\delta 1_M$) in the playback rate adaptation and in the average idle percentage. Although we consider this a minor drawback as it occurs only in transactional periods and only temporarily decreases the playback rate 0-10% than required there is a need for a more advanced DBTS in order mitigate this effect. We are currently working in this issue but we consider it is outside the scope of this paper as PRC is an independent component.

VIII. RELATED WORK

As observed a P2P overlay can act as a very helpful substrate in bandwidth monitor and control systems for stable P2P live streaming if it is efficient and fair. With the term efficiencyis depicted its ability to exploit in a high degree the available bandwidth of the heterogeneous peers that participate in it. With the term fair isdepicted its ability to balance the percentage of the upload bandwidth of participating peers that is used while simultaneously it balances the percentage of the reception of the stream of participating peer.

Initially, research community focused on the development of advanced P2P overlay architectures in order to increase the efficiency of P2P live streaming systems. Several P2P overlay architectures as [1], [9] and [14] have been proposed which dynamically optimize the connections of each participating peer according to its upload bandwidth while in **Error! Reference source not found.** it is studied the structure of the P2P overlay graph in order to maximize peer cooperation. These architectures achieve the efficient exploitation of the upload bandwidth of each participating peer even in highly heterogeneous environments.

Besides the efficient exploitation of heterogeneous upload bandwidths [2], [3] and **Error! Reference source not found.** propose P2P overlays that additionally achieve to minimize the inter ISP traffic that P2P live streaming introduces by connecting in the P2P overlay peers that are close to the underlying network.

A final objective to the design of P2P overlays is to enforce fairness. In [35] it is studied the tradeoff between fairness and efficiency by recognizing it as a very complex issue. On the other hand, although complete fairness in a distributed and dynamic way isn't an easy objective [2], [3] and [35] did progress towards this goal.

Despite their success P2P overlays are not able to guarantee QoS (successful stream delivery) under the lack of upload bandwidth which consequently leads to waste idle upload bandwidth in case that there is surplus.

In order to mitigate these effects research community proposed architectures that multiplex the upload bandwidth that is available in various P2P overlays that distribute various streams. In [24] it is exploited optimization theory in order to solve the problem of peer bandwidth allocation to multiple P2P overlays while [26] proposes collaborative optimization by decoupling the upload stream process from the stream consumption process. Each peer may upload part of a different stream than the one which consumes. In this way these approaches achieve the collaborative distribution of all the streams in case that the sum of the surplus bandwidth, in P2P overlays in which it appears, is greater than the sum of the deficit of the upload bandwidth, in P2P overlays in which it appears. Although, still these architectures fail, in case that bandwidth demands for the distribution of all the streams are greater that the bandwidth that is offered from all participating peers in all P2P overlays.

There are two strategies in order to adapt the peer to peer live streaming to the dynamic upload bandwidth of participating peers and ensure QoS in any case. The first is to dynamically allocate upload bandwidth to increase the upload capacity by using helpers [29],[30]. The helper's role can be played by idle peers [31] or dedicated servers [32],[33] from the cloud. The second is to adapt the playback rate according to the existing upload bandwidth. The selection of a strategy corelated with the QoS that users desire and the business model of the service provider and is not a pure technical decision. In case that users and service provider desire a costless live steaming the second strategy has to be selected. In case that desired a live streaming with very high QoS the first strategy has to be selected.

Towards the first strategy, the research community has proposed monitoring systems such as [5], [8] that use stochastic methods for dynamic and scalable monitor of total available bandwidth resources existing in a P2P overlay and calculate the deficit or surplus of the respective total upload bandwidth in order to furtherly allocate it. These systems are scalable but suffer from three drawbacks. The first is that stochastic methods are suitable only for specific upload bandwidth distributions among participating peers. The second is that, due to their low confidence interval, their efficiency in terms of peer bandwidth exploitation may be low. The third is that they are not capture the system dynamics and they don't offer QoS in cases of sudden disturbances (i.e. underlying network traffic changes and/or peers' arrivals-departures.

In **Error! Reference source not found.** the formation of a monitoring tree that aggregates monitoring data from participating peers provides a scalable solution in order to calculate dynamically the amount of existing resources. Furthermore, a dissemination protocol allows the dynamic configuration of the amount of cloud resources that the P2Poverlay needs. Also [34] considers a replicated service on top of a mixed P2P and cloud system. This protocol is able to self-regulate the amount of cloud storage resources utilization according to available bandwidth resources. As analyzed in [36] another requirement from a QoS strategy is to protect the system from malicious peers while also locality has to be taken into consideration as in **Error! Reference source not found.**. Although these systems are scalable and more efficient, they do not capture the system dynamics and so they are vulnerable to adverse changes of the upload bandwidth which could be consideredas lack of stability.

In **Error! Reference source not found.** the problem of stability is recognized and studied in detail the impact of flash crowds on the stability of the distribution of the stream. Moreover, is described in which cases admission control is mandatory to be used. In **Error! Reference source not found.** is studied the stability of a real P2P live streaming system called UUSee [23] and is highlighted that the server plays an indispensable role in the stability.

In our work we concern about the second QoS strategy which is dynamic playback rate adaptation. We develop a P2P live streaming system that is able to dynamically monitor with very low overhead the conditions of bandwidth resources in a P2P overlay and to ensure the complete (stable) distribution of the stream even in very adverse bandwidth changes. To the best of our knowledge this is the first work that achieves these two objectives.

IX. CONCLUSIONS

In this work presented a system that is able to monitor dynamically, in a scalable way, the upload bandwidth resources of participating peers in a P2P live streaming system and to adapt dynamically the playback rate of the video stream according to the aforementioned resources. Our evaluation testifies the existence of a system that efficiently exploits the available resources and delivers successfully the video stream even in adverse network conditions. For our future work we aim to evolve our system in order not only to dynamically adapt the video playback rate but also dynamically exploiting bandwidth from clouds and/or other external sources, in cases that it needed, for the stable distribution of the stream with increased QoS.

ACKNOWLEDGMENTS

This has been co-financed by STEER [10] which is an European Commission’s Seventh Framework Programme (FP7 2007-2013) under grant agreement nº 318343 that our university coordinates and by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

REFERENCES

[1] Fabio Picconi, Laurent Massoulie (2008) Is there a future for mesh-based live video streaming? In: Proceedings of the 8th Int. Conf. Peer-to-Peer Computing. IEEE, pp. 289 - 298

[2] Efthymiopoulos N, Christakidis A, Denazis S, Koufopavlou O (2011) Liquidstream – network dependent dynamic P2P live streaming. Springer Peer-to-Peer Networking and Applications 4(1):50-62

[3] Christakidis A, Efthymiopoulos N, Fiedler J, Dempsey S, Koutsopoulos K, Denazis S, Tombros S, Garvey S,Koufopavlou O (2011) VITAL++ a new communication paradigm: embedding P2P technology in next generation networks. IEEE Communications Magazine 49(1):84-91

[4] Yang Y, A L H Chow, L Golubchik, D. Bragg (2010) Improving QoS in BitTorrent-like VoD Systems. In: Proceedings of IEEE INFOCOM. IEEE, pp.1-9

[5] Delia Ciullo, Valentina Martina, Michele Garetto, Emilio Leonardi, Giovanni Luca Torrisi (2012) Stochastic Analysis of Self-Sustainability in Peer-Assisted VoD Systems. In: Proceedings of IEEE INFOCOM. IEEE, pp. 1539-1547

[6] PPLive, [Online]. Available: http://www.gridcast.cn

[7] Cisco visual networking index: Global mobile data traffic forecast upyear, 2010-2015 (2011) White Paper, Cisco

[8] Delia Ciullo, Valentina Martina, Michele Garetto, Emilio Leonardi, Giovanni Luca Torrisi (2012) Performance Analysis of Non-stationary Peer-assisted VoD Systems. In: Proceedings of IEEE INFOCOM. IEEE, pp. 3001-3005

[9] R Birke, E Leonardi, M Mellia, A Bakay, T Szemethy, C Kiraly, R Lo Cigno, F Mathieu, L Muscariello, S Niccolini, J Seedorf, G Tropea (2011) Architecture of a network-aware P2P-TV application: the NAPA-WINE approach. IEEE Communications Magazine 49(6):154-163

[10] Social Telemedia Environment for Experimental Research, [Online]. Available: http://www.steer.ics.ece.upatras.gr

[11] Opnet modeler, [Online]. Available: www.opnet.com

[12] Chi-Tsong Chen (2012) Linear System Theory and Design. Oxford Univeristy Press, USA

[13] Jean-Jacque Slotine, Weiping Li (1991) Applied Nonlinear Control. Prentice Hall

[14] A Couto da Silva, E Leonardi, M Mellia, M Meo (2011) Exploiting Heterogeneity in P2P Video Streaming. IEEE Transactions on Computers 60(5):667-679

[15] Jean-Jacque Slotine, Weiping Li (1991) Applied Nonlinear Control. Prentice Hall

[16] GridCast, [Online]. Available: http://www.gridcast.cn

[17] PPStream, [Online]. Available: http://www.ppstream.com

[18] TVU, [Online]. Available: http://www.tvunetworks.com

[19] SopCast, [Online]. Available: http://www.sopcast.com

[20] Y Chen, B Zhang, C Chen, D M Chiu (2014) Performance modeling and evaluation of peer-to-peer live streaming systems under flash crowds. IEEE/ACM Trans Netw 22(4):1106–1120

[21] N Magharei, R Rejaie, I Rimac, V Hilt, M Hofmann (2014) Isp-friendly live p2p streaming. IEEE/ACM Trans Netw 22(1):244–256

[22] C Wu, B Li, S Zhao (2012) Diagnosing Network-wide P2P Live Streaming Inefficiencies. ACM Trans multimedia Comput. Commun. Appl. 8(1S):Article 13

[23] UUSee, [Online]. Available: http://www.uusee.com

[24] Glauber D Gonçalves, Ítalo Cunha, Alex B Vieira, Jussara M Almeida (2014) Predicting the level of corporation in a Peer-to-Peer live streaming application.Springer Multimedia Systems pp.1-20

[25] Jiayi Liu, Shakeel Ahmad, Eliya Buyukkaya, Raouf Hamzaoui, Gwendal Simon (2014) Resource allocation in underprovisioned multioverlay peer-to-peer live video sharing services.Springer Peer-to-Peer Netw. Appl. pp.1-15

[26] D Wu, C Liang, Y Liu, K. Ross (2009) View-Upload Decoupling: A Redesign of Multi-Channel P2P Video Systems. In: Proceedings of IEEE INFOCOM. IEEE, pp. 2726-2730

[27] A Payberah, H Kavalionak, V Kumaresan, A Montresor, S Haridi (2012) Clive: Cloud-Assisted p2p Live Streaming. In: Proceedings of 12th Int. Conf. Peer-to-Peer Computing. IEEE, pp. 79–90

[28] D Frey, R Guerraoui, A Kermarrec, M Monod (2010) Boosting Gossip for Live Streaming. In: Proceedings of the IEEE 10th Int. Conf. on Peer-to-Peer Comp. IEEE, pp. 1–10

[29] J Wang, K Ramchandran (2008) Enhancing Peer-to-Peer live multicast quality using helpers. In: Proceedings of the IEEE 15th Int. Conf. on Image Processing. IEEE, pp. 2300-2303

[30] H Zhang, J Wang, M Chen, K Ramchandran (2009) Scaling Peer-to- Peer Video-on-Demand Systems Using Helpers. In:Proceedings of the IEEE 16th Int. Conf. on Image Processing. IEEE, pp. 3053-3056

[31] J Wang, C Yeo, V Prabhakaran, K Ramchandran (2007) On the role of helpers in Peer-to-Peer file download systems: Design, analysis and simulation. In: Proceedings of the 6th Int. Workshop on Peer-To-Peer Syst. (IPTPS 2007)

[32] R Sweha, V Ishakian, A Bestavros (2012) Angelcast: cloud-based peerassisted live streaming using optimized multi-tree construction. In: Proceedings of the 3rd Multimedia Systems Conference. ACM, pp. 191–202

[33] P Michiardi, D Carra, F Albanese, A Bestavros (2012) Peer-assisted content distribution on a budget. ELSEVIER Computer Networks 56(7):2038–2048

[34] H Kavalionak ,A Montresor (2012) P2P and cloud: A marriage of convenience for replica management. In: Proceedings of the 6th IFIP Int. Workshop on Self-Organizing Systems, ser. Lecture Notes in Computer Science. Springer, pp. 60–71

[35] D Wu, Y Liang, J He, X Hei (2013) Balancing Performance and Fairness in P2P Live Video Systems.IEEE Circuit and Systems for Video Techology 23(6):1029 - 1039

[36] Y Hu, D Dong, J Li, F Wu (2013) Efficient and incentive-compatible resource allocation mechanism for P2P-assisted content delivery systems. ELSEVIER Future Generation Computer Systems 29(6):1611-1620

[37] J Zhao, C Wu, X Lin (2013) Locality-aware streaming in hybrid p2p-cloud cdn systems. Springer Peer-to-Peer Networking and Applications 8(2): 320-335