

QOS - LIQUIDSTREAM: SCALABLE MONITORING AND BANDWIDTH CONTROL IN PEER TO PEER (P2P) LIVE STREAMING

Nikolaos Efthymiopoulos², Maria Efthymiopoulou³, Athanasios Christakidis¹

¹Department of Electrical and Computer Engineering, University of Patras, Patras, Greece

²National Technical University of Athens, Athens, Greece

³Tei of Western Greece, Antirio, Greece

ABSTRACT

The vast majority of research in P2P live streaming systems focuses on system architectures that offer to participating peers: high upload bandwidth utilization, low delays during the video stream diffusion, robustness and stability under dynamic network conditions and peers behavior. On the other hand in order to guarantee the complete and on time video distribution to every participating peer, the average upload bandwidth of the participating peers should be always greater than the playback rate of the video stream. Most of the approaches do not take into consideration this requirement. Thus, in this paper we propose a very scalable monitoring mechanism of the total upload bandwidth of the participating peers, which is dynamic, accurate and with low overhead. Moreover, by exploiting this monitoring mechanism we present and evaluate an algorithm that allows the accurate and on time estimation of the minimal required additional bandwidth that an external set of resources (e.g. auxiliary peers) have to contribute. In this way we guarantee the uninterrupted the stream delivery and provide high Quality of Service (QoS) in live streaming.

KEYWORDS

peer to peer (P2P), live streaming, scalable monitoring, bandwidth allocation

1. INTRODUCTION

P2P live streaming has been a thoroughly researched topic with many contributions from the scientific community [1],[2],[4],[8],[10],[11],[13],[14],[23],[24],[25] that lead to many commercial systems such as [16],[17],[18],[19]. The main objective of these systems is the uninterrupted stream delivery with high QoS and Quality of Experience (QoE) with respect to conventional content distribution systems.

A P2P live streaming system, in order to be effective, attractive and competitive with traditional approaches, should meet the following requirements. These are:

- To fully utilize the heterogeneous upload bandwidth of participating peers. The sum of the upload bandwidth of the participating peers constitutes the core of the system's resources. Specifically, according to [2], each time instant the average upload bandwidth of the participating peers forms an upper limit to the video stream bit rate (playback rate) that any P2P system is able to deliver.

- To minimize this latency. We define as latency (setup time) the time interval between the generation of a block from its “source” (e.g. the video server) until its distribution to every participating peer.
- The fast adaptation to users behavior. Due to peers churn the system must be able to adapt its flows dynamically and efficiently according to existing resources.
- Uniform distribution of the total upload bandwidth among the requests of participating peers. In case of sufficient average upload bandwidth the system should be able to ensure that every peer will eventually acquire all video blocks. Whereas, if the average upload bandwidth becomes less than the bit rate of the video stream, then an efficient system must deliver the same percentage of video blocks to every participating peer.

From ISPs perspective, the goal is to minimize the cost for the content distribution per peer, without compromising QoS. This depends on the number of video servers and caching centers that have to be deployed and maintained. . For these reasons, the use of P2P approach is an alternative or complementary mechanism to mainstream content distribution architectures. In this way we also keep low the underlying network traffic that participating peers generate.

Referring to system's implementation, there are effective P2P overlay algorithms [8],[21],[22],[26] that lead to upload bandwidth utilization more than 90%, while setup time in a live streaming system has been brought down to 5-10 sec or even less [11]. Another issue in P2P live streaming is to create architectures that are able to adapt the block transmissions dynamically, according to the set of the blocks that peers miss in each time instant. Recently proposed P2P block exchange scheduling architectures (e.g. [10]) try to handle this issue.

Nevertheless, when these systems are deployed in real conditions, as commercial or operational P2P live streaming systems [16],[17],[18],[19], they fail to achieve the reported performance and stability. Even an inexperienced user can notice high latencies, instability and low bit rates of the delivered video stream.

This is mainly attributed to the fact that the average upload bandwidth of the participating peers changes unpredictably over time, according to the number of participating peers and the network conditions that might occur at each time instant. As a result, the designers of these systems act conservatively and select low video stream bit rates for achieving high probability of complete stream delivery. This explains low video bit rates that users experience in commercial P2P live streaming systems. It is noticeable that when the average peers upload bandwidth becomes lower than the video bit rate, there is lack of algorithms that could handle system's instability. This disturbance, due to missed video blocks, leads to low QoE. Recently, the research community focused on video coding algorithms and information models that are able to adapt the required resources for the video distribution to the available bandwidth resources of the participating peers. However, these efforts still suffer from low QoS for the sake of stability.

This paper focuses on a fundamental problem: the real time, dynamic, scalable and accurate estimation of the total bandwidth resources that each P2P live streaming system offers. This could be achieved by calculating the resources that are missing or over provisioned. Moreover, in this paper is proposed an architecture that allocates and embeds in the P2P overlay, in an efficient way, auxiliary peers for the efficacious and stable video stream distribution. The innovation of our approach is highlighted through three steps.

Firstly, is proposed a novel scalable architecture able to monitoring in real time, dynamically and accurately the average upload bandwidth of participating peers in a video distribution by introducing negligible overhead to the system. The output of this algorithm is the estimation of the total bandwidth resources that each video distribution has.

Secondly, is proposed an algorithm that by using as input the output of the aforementioned monitoring algorithm, allocates dynamically (from a set of auxiliary peers) the minimal required bandwidth resources towards the complete distribution of the video stream (in case of bandwidth deficit). This algorithm uses as input: a) uploading capabilities of a set of nodes (peers and/or servers) denoted as auxiliary peers, b) the total amount of the upload bandwidth of the participating (as this is estimated from the monitoring mechanism), and c) the amount of resources that the distribution requires. In this way, it calculates the amount of bandwidth that each auxiliary peer has to contribute. The objective of this algorithm is to allow auxiliary peers to share the workload dynamically and according to their bandwidth capacity.

Thirdly, the goal is to further extend our proposed architecture in order to involve dynamically auxiliary peers in the system and exploit the allocated resources in a distributed and dynamic fashion. In order to achieve this, we exploit the overlay that is presented in [2] which is adaptable to dynamic bandwidth changes and suitable for environments where peers have high levels of heterogeneity (in terms of their upload bandwidth capabilities) This overlay is also adaptable to the underlying network topology and traffic.

This work is based on our previous work [2]. By exploiting the aforementioned desirable properties of our proposed P2P overlay and block scheduler we achieve the efficient functionality of monitoring and bandwidth control architecture. The motivation behind our work and more information about the deployment of content distribution services, by using P2P architecture, is described in [2],[27].

The remaining of the paper is structured as follows. In section 2 we formulate the problem. In section 3 we describe our previous work that is used as background knowledge in this paper. In section 4 we present the innovative monitoring mechanism and its features. In section 5 we describe the required upload bandwidth resources estimation process and we propose a policy for balanced resource allocation among auxiliary peers. In section 6 we demonstrate how auxiliary peers are dynamically organized in the video stream diffusion graph. In section 7 the evaluation of our system is presented. Finally, section 8 presents the conclusions of this paper and the future work.

2. PROBLEM STATEMENT

Every P2P live streaming system involves at every time instant, t , a set of participating peers, noted as $S(t)$, where each one of them contributes to the system its upload bandwidth, $c_i(t), i \in S(t)$. We also assume that at any time instant there is another non empty set, noted as $S_{aux}(t)$, of *auxiliary* peers that can become available upon request by contributing certain amounts of upload bandwidth, $c_{aux,i}(t), i \in S_{aux}(t)$.

These auxiliary peers could be servers from a service provider, caches in the network or regular users that participate in other video stream distributions and have idle resources. They may be used to support various service models such as: server assisted P2P live streaming, multi-channel P2P live streaming and network cache assisted P2P live streaming.

As auxiliary resources cost, sharing them across a number of overlays in an optimal way is of paramount importance. To this end, by minimizing the amount of auxiliary resources that should be allocated to a specific stream distribution without compromising stability and efficiency, maximizes CAPEX/OPEX. Assume that a P2P live streaming system distributes a video with playback rate of μ bps. If each auxiliary peer has to contribute upload bandwidth $c_{aux,i}(t)$ at any time instant, the goal is to minimize the following aggregate function that represents the total amount of upload bandwidth contributed by the set $S_{aux}(t)$. Thus:

$$\min \sum_{i \in S_{aux}(t)} c_{aux,i}(t) \quad (1)$$

We define as $f(i,j)$ the bandwidth upload rate of peer i that directed to peer j . In order to guarantee the complete and on time stream delivery to every participating peer the following three equations must hold.

$$\sum_{i \in S(t) \cup S_{aux}(t)} f(i,j) \geq \mu, \forall j \in S(t) \cup S_{aux}(t) \quad (2)$$

$$\sum_{j \in S(t) \cup S_{aux}(t)} f(i,j) \leq c_i(t), \forall i \in S(t) \quad (3)$$

$$\sum_{j \in S(t) \cup S_{aux}(t)} f(i,j) \leq c_{aux,i}(t), \forall i \in S_{aux}(t) \quad (4)$$

Eq. 2 expresses that the total incoming rate to a peer j must be equal or higher than the playback rate. While Eq. 3 and Eq. 4 represent that the total outgoing rate from every peer will be lower or equal than its capacity. Assume now that N is the cardinal number of $S(t) \cup S_{aux}(t)$.

By summing up Eq. 2 for each participating peer (N equations), we have the total incoming flow that the system generates. In addition, by summing up Eq. 3 and Eq. 4 for "consumers" and auxiliary peers respectively, we have the total outgoing flow that the system generates (N equations).

In every P2P live streaming system the outgoing flow is equal with the incoming flow. So, by subtracting the N equations of type (3) and (4) from the N equations of type (2) we have:

$$\sum_{i \in S_{aux}(t)} c_{aux,i}(t) \geq N * \mu - \sum_{i \in S(t)} c_i(t) \quad (5)$$

Let examine Eq. 1 and Eq. 5 conjointly. The equality in Eq. 5 corresponds to the minimal amount of bandwidth that auxiliary peers have to contribute for the successful delivery of the stream.

In order to calculate and allocate the amount of bandwidth that each auxiliary peer has to contribute, we need to accurately estimate the total amount of upload bandwidth resources that contributed by the participating peers, $S(t)$. The periodic measurement of every participating peer upload bandwidth is a non-scalable process and doesn't capture bandwidth fluctuations. In order to meet these requirements (scalability, accuracy and real time operation) there is a need for a measurement process that relies on statistical sampling. In section IV is described such a process that estimates the total upload bandwidth capabilities of the participating peers by exploiting the properties of P2P block schedulers.

After the estimation of the total upload bandwidth that participating peers contribute (through Eq. 5) we are able to calculate the total upload bandwidth that auxiliary peers ($S_{aux}(t)$) have to contribute. As a next step, we could apply a bandwidth allocation algorithm that apportions this amount of bandwidth to the set of auxiliary peers that belong to $|S_{aux}(t)|$ in order to have equal percentages of bandwidth utilization among them. The bandwidth allocation algorithm will result in two possible outcomes. The first is the allocation of more bandwidth through auxiliary peers (in case that the average uploading bandwidth of participating peers decreases). The second is the reduction of the bandwidth that the set of auxiliary peers already contributes (in case that the average upload bandwidth of the participating peers, that belonging to $S(t)$, increases).

Finally, an efficient combination of a P2P overlay algorithm and scheduler that readily reacts to peers churn and adapts to bandwidth changes is necessary. Such a system is part of our previous work and is briefly presented in the next section.

3. P2P LIVE STREAMING

In [2] we focused on the development of a P2P live streaming system that consists of two inter-related entities: the Content Diffusion Overlay (CDO) and the P2P Block Exchange Scheduler (BESA).

3.1. CONTENT DIFFUSION OVERLAY

A P2P overlay architecture, suitable for P2P live streaming system, should meet the following requirements. Firstly, the overlay graph should be constructed in such a way that every peer will have a sufficient number of neighbors proportional to its uploading bandwidth. This guarantees optimal upload bandwidth utilization, which in turn has a positive impact on block scheduling. Likewise, each peer should have a sufficient number of incoming connections for the undistruptive reception of the video stream regardless of the dynamic network conditions and/or peer arrivals and departures. In addition, the overlay should be dynamically reconfigurable in order to dynamically react to the various changes of the underlying network as well as to the dynamic behavior of the participating peers. Last but not least, it should exploit the underlying network latencies (i.e. round trip time between peers) which means that each peer should have as its neighbors those peers that are close to him in the network (we refer here to the network distance). In other words, the overlay must reflect as much as possible locality information in the way that peers are kept organized. Our proposed P2P overlay architecture derived from the aforementioned requirements (Figure 1).

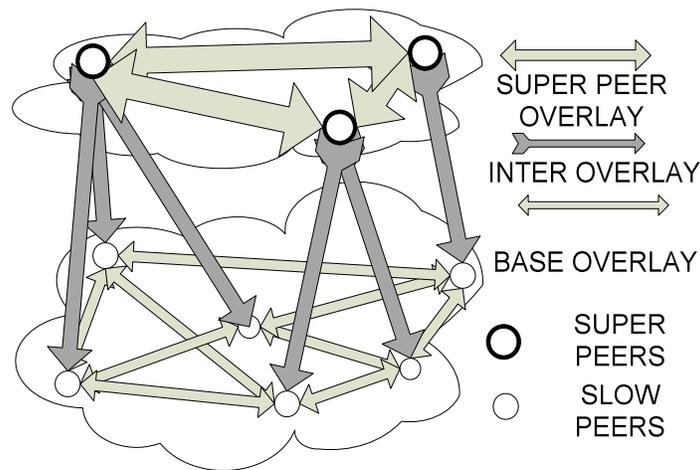


Figure 1. The structure of the overlay graph

We distinguish between two types of peers: the super peers and the slow peers. The former are those with upload bandwidth higher than the video playback rate, whereas the latter are peers with upload bandwidth less than the video playback rate.

Every peer that joins the system it becomes part of a *base overlay* and is assigned to it a fixed number of neighbors (M_B). This is a bidirectional mesh overlay, balanced with respect to the number of neighbors. If this peer also happens to be a super peer then it is also admitted to an additional overlay, called *super-peer overlay*, of similar characteristics with the base overlay. In super-peer overlay, the peer is also assigned a fixed number of neighbors, say M_S .

Finally, the *inter overlay* connects the base and super-peer overlays by assigning a number of super peers to each slow peer. More specifically, each slow peer in the base overlay selects a

fixed number of super peers, M_s , which wishes to connect with. These interconnections are unidirectional, originating from the super peers (outgoing reconnections) and arriving to the slow peers (incoming connections). Interconnections are also distributed among super peers in a manner proportional to the excess of their uploading bandwidth (uploading bandwidth minus the video playback rate). The quantities M_B , M_S and M_I are parameters of the system overlay. For instance, Figure 1 depicts a P2P overlay with $M_B=3$, $M_S=2$, and $M_I=1$.

All the peers periodically execute a Distributed Optimization and Maintenance Algorithm (DOMA) that reorganizes the “neighborhoods” of the CDO in order to keep the structure of the graph optimal for the content delivery, even during peer arrivals and departures. The optimization algorithm makes use of an “energy function” that captures the impact of specific parameters such as network latency among peers. In this way are created locality aware overlays.

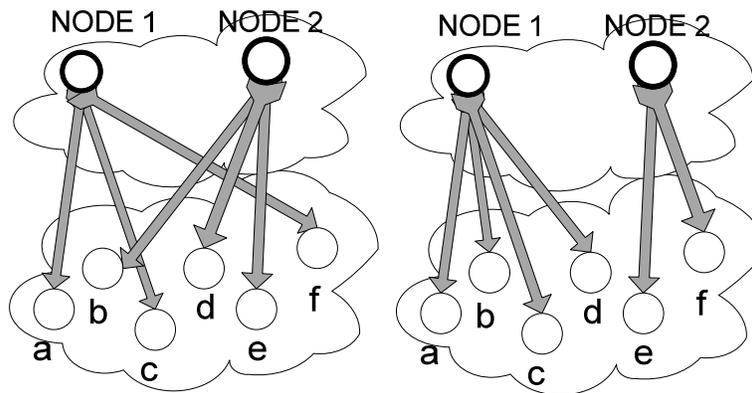


Figure 2. Execution of DOMA

Changes in the underlying network conditions (that change the amount of resources that peers contribute) or peers arrivals/departures are mirrored in DOMA execution. We stress here that DOMA always converges to the desired graph structure and to a minimized sum of energies [2].

3.2.P2P BLOCK EXCHANGE SCHEDULING ALGORITHM

In conjunction with DOMA that runs in every participating peer, a P2P Block Exchange Scheduler Algorithm (BESA) ensures the distribution of each video block to every participating peer in the CDO. Our BESA [2] has been designed in such a way that exploits CDO's structure and properties (e.g. locality). Moreover, referring to its execution is a periodic process by each participating peer. At each time interval every peer acts as a sender and receiver simultaneously. This is done through the use of the corresponding BESA's decision functions.

In order to achieve fair block distribution and build a system adaptive to dynamic peers behavior and network conditions, the selection of the receiving peer is the responsibility of the sending peer. This selection is taking place exactly before the beginning of the transmission of a video block. In order to facilitate this process the receiving peers proactively notify candidate sending peers about the blocks that they miss and wish to receive. In this way we achieve fast and complete diffusion of video blocks. Additionally duplicate block transmissions from different sending peers to the same receiving peer are avoided. We consider this receiver driven block selection approach as the most efficient one in distributing video blocks. The receiver peer has always a better knowledge about the rarity of its missing blocks, that whereas exist in the buffers of its neighbors and this knowledge is communicated with its neighbors (that will act as sending peers). In this way, the problem of video block distribution has been shifted towards coordinating

these sending peers in a distributed manner. Consequently, duplicate block transmissions are avoided and the transmission of rare blocks in every neighborhood is prioritized.

Furthermore, as the rate of receiving peer's requests has to be kept at least equal to the video playback rate, the receiving peer sends its block requests only to those candidate sending peers that can meet this constraint. Namely, they have sufficient upload bandwidth. Towards this goal, each peer implicitly announces its serving capability by periodically issuing tokens to a number of peers, with size equal to its upload bandwidth. These tokens have to be distributed uniformly to the participating peers in order to request blocks and eventually acquire the video stream.

BESA achieves high bandwidth utilization, fast diffusion of each video block, uniform distribution of the total upload bandwidth resources to the participating peers and low control overhead. We exploit these properties towards the development of our monitoring and resource allocation algorithms.

4. P2P MEASUREMENTS

This section describes a measurement process that constantly estimates the total upload bandwidth of a P2P live streaming system. This could be achieved by using only a small subset of the participating peers, in which is measured periodically their incoming bit flows during a time period, named T_A . As a next step, these measurements are propagated to a server that periodically executes an algorithm in order to estimate system's total upload bandwidth.

During every time interval, T_A , each peer i could be either busy, by transmitting for a period, $T_{busy}(i)$, or idle for a period $T_{idle}(i)$. Thus, $T_A = T_{busy}(i) + T_{idle}(i)$. Accordingly, every peer i could estimate its upload bandwidth $c_i(T_A)$ as the ratio of the transmitted bits, $B_{out}(i)$, over the time interval $T_{busy}(i)$. More information on this measurement process can be found in [7] Summing up all these measurements carried out by peers, we can express the total upload bandwidth of every set of participating peers, namely,

$$\sum_{\forall i} c_i(T_A) = \sum_{\forall i} \left[\frac{B_{out}(i)}{T_{busy}(i)} \right] \quad (6)$$

or, alternatively,

$$\sum_{\forall i} c_i(T_A) = \sum_{\forall i} \left[\frac{B_{out}(i)}{T_A - T_{idle}(i)} \right] \quad (7)$$

In order to simplify the sum in Eq. 7, we will exploit a property of the proposed BESA as described earlier. According to this property, due to the uniform exploitation of the participating peers upload bandwidth, the time interval, $T_{idle}(i)$, is almost the same for every participating peer. More precisely, by executing offline the Kolmogorov – Smirnov test [27] in the values of this metric among all participating peers, is testified that it approximates a normal distribution.

In order to be able to calculate the average values of the parameters above, we resort to sampling by selecting a small subset of peers, S_{sample} , making sure that the selected size of the subset gives accurate estimates of the averages.

Let us assume temporarily that a measurement server has already selected the member peers of such a set, S_{sample} , and requests from them the values of their $T_{idle}(i)$, which measured during the last period, T_A . The statistical average $E[T_{idle}]$ and the statistical standard deviation $\sigma_{T_{idle}}$ can be calculated according to the following equations:

$$E[T_{idle}] = \frac{\sum_{i \in S_{sample}} T_{idle}(i)}{|S_{sample}|} \quad (8)$$

$$\sigma_{T_{idle}} = \sqrt{\sum_{i=1}^{|S_{sample}|} \frac{(T_{idle}(i) - \overline{T_{idle}})^2}{|S_{sample}| - 1}} \quad (9)$$

With probability α , the value $T_{idle}(i)$ of a participating peer i lies in the following interval:

$$E[T_{idle}] \mp z_{\alpha/2} * \sigma_{T_{idle}} / \sqrt{|S_{sample}|} \quad (10)$$

Where $z_{\alpha/2}$ is the value of a random variable that follows a normal distribution for which the integral of the probability density function of the normal distribution is $\alpha\%$. We highlight here two facts. The first is the very low statistical standard deviation of this metric, as our BESA exploits uniformly the upload bandwidth of participating peers. The second is that the value of T_{Δ} is much higher than $T_{idle}(i)$. The goal is to minimize the upload bandwidth that auxiliary peers have to contribute. Thus, $T_{idle}(i) \ll T_{\Delta} = T_{busy}(i) + T_{idle}(i)$. Consequently, is introduced an error smaller than:

$$z_{\alpha/2} * \sigma_{T_{idle}} / \sqrt{|S_{sample}|} \quad (11)$$

In this way, according to Eq. 11, Eq. 7 could be expressed as:

$$\sum_{\forall i} c_i(T_{\Delta}) = \frac{\sum_{\forall i} B_{out}(i)}{T_{\Delta} - T_{idle}} \quad (12)$$

In every P2P live streaming system, at a given time interval, the sum of the outgoing flows is identical with the sum of the incoming flows. Roughly speaking, the total amount of outgoing bits is identical equal with the total amount of incoming bits. According to this fact Eq. 12 could be rewritten as:

$$\sum_{\forall i} c_i(T_{\Delta}) = \frac{\sum_{\forall i} B_{in}(i)}{T_{\Delta} - T_{idle}} \quad (13)$$

The statistical average $E[B_{in}]$ and the statistical standard deviation $\sigma_{B_{in}}$ are calculated according to the following equations:

$$E[B_{in}] = \frac{\sum_{i \in S_{sample}} B_{in}(i)}{|S_{sample}|} \quad (14)$$

$$\sigma_{B_{in}} = \sqrt{\sum_{i=1}^{|S_{sample}|} \frac{(B_{in}(i) - \overline{B_{in}})^2}{|S_{sample}| - 1}} \quad (15)$$

By applying to these measurements the Kolmogorov – Smirnov test [7], it could be evaluated that the probability density function of various $B_{in}(i)$ approximates the normal distribution. By modeling this distribution as the normal distribution, it could be estimated the average value of B_{in} with a confidence interval α according to:

$$P \left[-z_{\frac{\alpha}{2}} \leq \frac{E[B_{in}] - \sigma_{B_{in}}}{\frac{\sigma_{B_{in}}}{\sqrt{|S_{sample}|}}} \leq z_{\frac{\alpha}{2}} \right] = 1 - \alpha \quad (16)$$

Eq. 16 states that for every parameter that follows the normal distribution the mean value is estimated in the following interval with probability α :

$$E[B_{in}] \mp z_{\alpha/2} * \sigma_{B_{in}} / \sqrt{|S_{sample}|} \quad (17)$$

By setting the parameter α very close to 100% (ex. $\alpha=0.99$) we estimate, with very high confidence and by measuring only a small subset of peers, the lower threshold of value $\overline{B_{in}}$ which is:

$$\overline{B_{in}} = E[B_{in}] - z_{\alpha/2} * \sigma_{B_{in}} / \sqrt{|S_{sample}|} \quad (18)$$

As could be observed from Eq. 18 the size of the interval, in which $\overline{B_{in}}$ could be estimated, depends on the size of the set S_{sample} and on the statistical standard deviation $\sigma_{B_{in}}$. This observation has a great impact on the statistical behavior that should be sought after from BESA. The lower the statistical standard deviation, the smaller the size of the S_{sample} that is required in order to estimate and calculate $\overline{B_{in}}$. To this end, our BESA, that achieves uniform block distribution among peers, results in small sample sizes. This will be further analyzed in the evaluation section. Exceeding a certain threshold has no effect in the accuracy of the measurements. Based on the exploitation of this observation, our measurement algorithm achieves scalability and accuracy.

The parameters in the right part of Eq. 7 depend on the index i (peer i). However, our measurement process is interested only for the average values. Taking therefore the corresponding statistical averages $\overline{B_{in}}$ and $\overline{T_{idle}}$ for every peer i , we have the following equation that describes the total upload bandwidth of a system comprised of a set of participating peers of size N :

$$\sum_{i=1}^N c_i(t) = \frac{N \times \overline{B_{in}}}{T_{\Delta} - \overline{T_{idle}}} \quad (19)$$

As stated in Eq. 2, the total amount of flows that each peer i has to acquire from the participating peers (for the complete reception of the video stream) is equal to the video playback rate, μ . However, it is noticed that P2P block schedulers introduce a control overhead during their operation due to network protocol headers, duplicate block transmissions, control messages that exchanged among peers etc.

This overhead ratio is denoted as r_{OHD} . Each peer can easily measure, at each T_{Δ} , its total incoming amount of bits, named as $\overline{B_{in}}$, and the amount of bits that are parts of the video blocks, named as $\overline{B_{in,actual}}$. Our measurement process can also estimate this overhead, r_{OHD} , by using the same peers in the S_{sample} . To this end, at every period, T_{Δ} , it could be estimated with accuracy the ratio r_{OHD} as:

$$r_{OHD} = \frac{\overline{B_{in}}}{\overline{B_{in,actual}}} \quad (20)$$

5. BANDWIDTH ESTIMATION AND RESOURCE ALLOCATION

In this section is analyzed the process of the additional bandwidth estimation that auxiliary peers have to contribute. This process is executed periodically, by using as input the measurements of the aforementioned monitoring mechanism. Moreover, is proposed an algorithm that calculates the bandwidth amount that each auxiliary peer has to contribute in order to balance the percentage of the bandwidth utilization among auxiliary peers. Through this policy the overloading a subset of auxiliary peers is avoided. In addition the percentage of idle resources, in each auxiliary peer,

are balanced. Accordingly, this amount of resources could be further shared among different overlays (e.g. multiple channel video distribution).

More specifically, we define as DU_{req} the difference between the minimum required bandwidth for the system (in order to ensure the complete and on time video distribution to every participating peer) and the total available bandwidth that the system has. In this way, Eq. 5 could be rewritten as:

$$DU_{req} = N * \mu - \sum_{i \in S(t) \cup Saux(t)} c(i) \quad (21)$$

Towards the formulation of this equation, in section III is assumed, for the sake of clarity, that P2P block schedulers introduce no overhead. Based on this fact, the total amount of bandwidth that is required for the complete distribution of a video stream with bit rate μ in N participating peers is $N * \mu$. However, by considering the overhead that introduced from the P2P block exchange scheduling algorithms, as analyzed in previous section, it obtained a more accurate estimation of the total required bandwidth for the complete and on time video distribution. Furthermore, the set $S(t) \cup Saux(t)$ is the sum of participating "consumers" and auxiliary peers. After these observations, Eq. 15 could be rewritten as:

$$DU_{req,actual} = N * \mu - \sum_{i \in N} c(i)_{actual} \quad (22)$$

The $DU_{req,actual}$ is the difference between the actual amount of bandwidth that auxiliary peers have to contribute and $\sum_{i \in N} c(i)_{actual}$ that is the actual total amount bandwidth of the participating peers. It is highlighted that with the term actual is noted the bandwidth that is really exploited, as it is excluded the overhead. This overhead is caused by a number of factors[27]. Among them are the network overhead (packet headers) and the overhead that P2P BESA introduces. Consequently, the ratio between the total bandwidth and the actual total bandwidth, without loss of generality, is equal to the overhead ratio. Thus, Eq. 22 could be rewritten as:

$$DU_{req} = \left(N * \mu - \frac{\sum_{i \in N} c(i)}{r_{OHD}} \right) * r_{OHD} \quad (23)$$

Moreover (by using Eq. 19 and Eq. 20) Eq. 22 could be rewritten as:

$$DU_{req} = N * \left(\frac{\overline{B_{in}}}{\overline{B_{in,actual}}} * \mu - \frac{\overline{B_{in}}}{\overline{T_{\Delta} - \overline{T_{idle}}}} \right) \quad (24)$$

By observing Eq. 23, in order to estimate the increase or decrease on the amount of bandwidth that auxiliary peers have to contribute, it is necessary only the estimation of $\overline{B_{in}}$, $\overline{B_{in,actual}}$ and $\overline{T_{idle}}$. As explained in the previous section, these metrics have been selected for estimation due to the fact that they have very low variance. Consequently, the measurement of these values in a small subset of participating peers is sufficient in order to estimate with accuracy system's total bandwidth.

The estimation of DU_{req} could lead in two different scenarios. According to the first one, DU_{req} will have a positive value. In this case, auxiliary peers have to contribute an amount of their upload bandwidth in order to allow the system to be able to ensure the complete and on time video distribution to every participating peer. According to the second scenario, DU_{req} will have negative value. In this case, the auxiliary peers upload bandwidth should be reduced, by a certain amount, in order to avoid the waste of network resources. This extra amount of bandwidth could be further used for other purposes (e.g. another object distribution). The usage of such an algorithm has two advantages. The first is that is achieved the contribution of the minimum

amount of bandwidth towards the complete distribution of the video stream. The second is that the system is able to accomplish dynamically the bandwidth requirements, by adapting the behavior of the auxiliary peers to the available bandwidth resources.

In this way is obtained the amount of bandwidth that auxiliary peers have to contribute in a rapid, accurate and scalable manner. Another problem that it has to be tackled is how to distribute DU_{req} among the set, $S_{aux}(t)$, of existing auxiliary peers. It is highlighted here that our system does not do any assumption on the size and the amount of bandwidth that each auxiliary peer contributes. Consequently, it could be capable for any P2P live streaming architecture.

The objective in sharing the DU_{req} among auxiliary peers is the balanced utilization of their uploading capabilities. We model here this problem as a convex network optimization problem. The analytical algorithmic solution of such kind of problems is further analyzed in [5]. As mentioned earlier, according to the first case, where DU_{req} is positive, the auxiliary peers have to contribute more upload bandwidth. Whereas, according to the opposite case, where DU_{req} is negative, the degradation of auxiliary peers bandwidth contribution is required.

In order to examine the first case, let us note as $c_{aux}(i)$ the total uploading bandwidth capacity of each auxiliary peer i that belongs to $S_{aux}(t)$. While $c_{auxi}(t)$ represents the upload bandwidth that auxiliary peer i currently contributes. In the same way $c_{auxi}(t+1)$ is the upload bandwidth that auxiliary i will contribute after the execution of the bandwidth allocation algorithm. In order to balance the utilization of the upload bandwidth of each auxiliary peer with equal percentage we model the reassignment of the upload bandwidth that auxiliary peers have to contribute as:

$$\min \sum_{i \in Sh(t)} \left(\frac{c_{hi}(t+1)}{c(i)} \right)^2 \quad (25)$$

A constraint that should be met is that the upload bandwidth that the bandwidth allocation algorithm could allocate from each auxiliary peer have to be less than its upload capacity. So for each auxiliary peer i the constraint is:

$$c_{auxi}(t+1) \leq c_{aux}(i) \quad (26)$$

Finally the supplement of bandwidth that $S_{aux}(t)$ will contribute will be:

$$\sum_{i \in S_{aux}(t)} [c_{auxi}(t+1) - c_{auxi}(t)] = DU_{req} \quad (27)$$

In case that the problem is infeasible, it means that the sum of the upload bandwidth that remains in $S_{aux}(t)$ is not enough for the complete and on time video distribution to every participating peer. In such case, the only solutions are either the reduction of the video stream bit rate, by using video coding techniques, or a lower quality of the video stream. We leave the solution of this problem as future work. In this work, we consider that always the total amount of bandwidth that is available from $S_{aux}(t)$ is enough for the delivery of the video stream.

We now examine the second case, where DU_{req} is negative, and so there is the opportunity to save upload bandwidth from the various auxiliary peers. The objective is to:

$$\min \sum_{i \in S_{aux}(t)} \left(\frac{c_{auxi}(t+1)}{c_{aux}(i)} \right)^2 \quad (28)$$

Under the constraint that:

$$\sum_{i \in S_{aux}(t)} [c_{auxi}(t + 1) - c_{auxi}(t)] = DUreq \tag{29}$$

In this case, the desirable for each auxiliary peer is:

$$c_{auxi}(t + 1) \geq 0 \tag{30}$$

In case where from these equations arises an infeasible problem, it means that there is no surplus of upload bandwidth in auxiliary peers set.

6. EMBEDDING AUXILIARY PEERS TO OVERLAY ARCHITECTURE

The set of the auxiliary peers (with their upload bandwidth contribution) that participate in the system is determined dynamically based on the output of the resource allocation algorithm. - Consequently, the CDO is adapting periodically and dynamically its structure based on these changes. For this reason, auxiliary peers are inserted as super peers in the super peer overlay in order to exploit our DOMA towards the utilization of the excess bandwidth that they provide. This could lead to fully exploitation of their upload bandwidth. Moreover, network traffic is minimized as these peers are connected to peers that are physically close to them in the underlying network. We again refer to [2] for the detailed description of DOMA that achieves this goal.

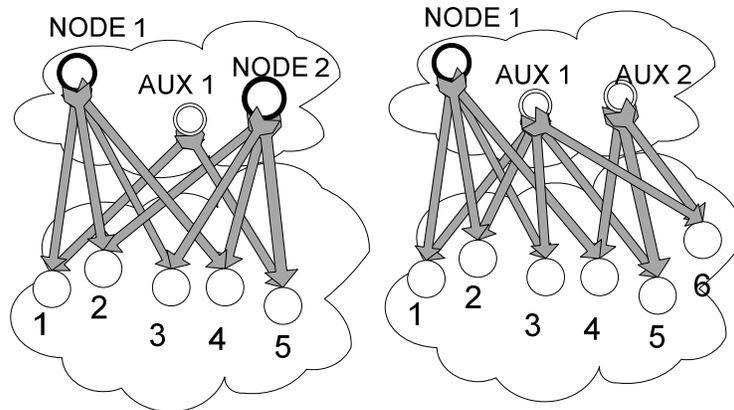


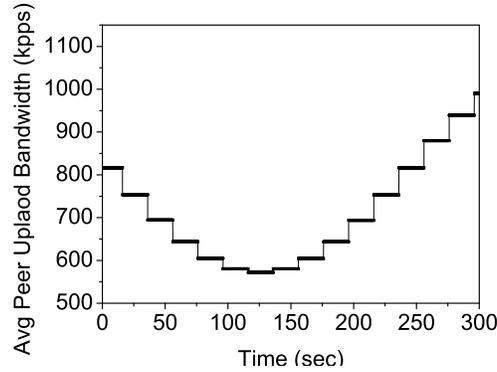
Figure 3. CDO adaptation based on the output of resource allocation algorithm

In Figure 3 is presented an example of a part of the CDO in two cases. In the first case (the left one), there are two super peers with equal upload bandwidth, named C , and an auxiliary peer that provides a quantity of $C/2$ of upload bandwidth. In the second case (the right one), one of the super peers has left the system and a new auxiliary peer entered to the system. Consequently, the resource allocation algorithm modified the bandwidth that auxiliary peers have to contribute to a quantity equal to $3/4 * C$ of upload bandwidth.

We highlight here that our distributed optimization algorithm ensure the uniform distribution of the connections between the two overlays, -according to super peers upload bandwidth. Additionally, these connections are uniformly distributed to slow peers according to the network latency between super and slow peers. In Figure 3 the length of the arc is proportional to the network latency among the connected peers.

7. EVALUATION

For the evaluation of our proposed P2P live streaming system is used the simulator OPNET Modeler v.14 [6] in which we conduct extensive simulations under various scenarios and conditions. Here we present its performance based on a network topology taken from [9]. In order to model heterogeneous uploading bandwidth capabilities of the participating peers we set four classes. Peers in class 4 have upload bandwidth equal to 4000 kbps, peers in class 3 have upload bandwidth equal to 1000 kbps, peers in class 2 have upload bandwidth equal to 384 kbps, and peers in class 1 have upload bandwidth equal to 128 kbps respectively. These uploading capacities are based on today's P2P live streaming systems trends.



Graph 1. Average upload bandwidth of participating peers through time

In this paper is demonstrated a scenario in which 500 peers participate in the system. The video playback rate that is distributed is $\mu=900$ kbps and the video has duration is 300 seconds. In Graph 1, is demonstrated the average uploading capacity of the participating peers over time. This scenario covers all possible cases. As could be observed from this graph, it includes cases where the average upload bandwidth of the participating peers is less than the required as well as cases vice versa. Additionally, discrete variations (steps) are included in order to make the problem more challenging.

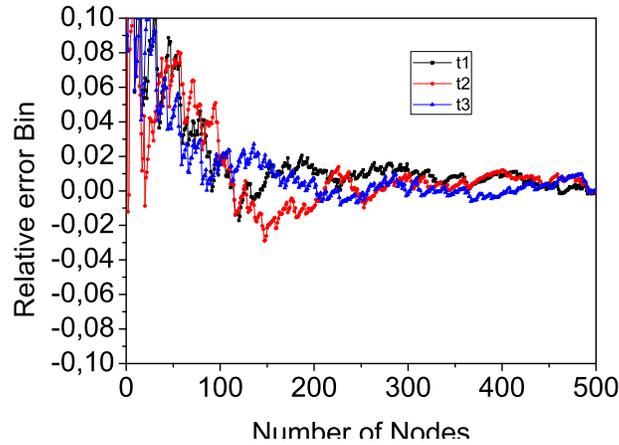
In order to demonstrate the accurate estimation of $\overline{B_{in}}$, by sampling only a small number of peers, we gathered the value of $\overline{B_{in}}$ from every participating peer in 20 random time instances. As a result, are presented three different time instances (t_1 , t_2 and t_3) in which the values of the measured B_{in} of the sampled peers exhibit the greatest variation. Through this presentation is depicted the way in which the estimation of $\overline{B_{in}}$ converges as the size of the set of the sampled peers increases. In more detail, in Graph 2 is demonstrated the relative error in the estimation of $\overline{B_{in}}$ over different sampling sizes. This can be calculated as:

$$RE(\overline{B_{in}(sample)}) = \left(\frac{\overline{B_{in}(sample)} - \overline{B_{in}(all)}}{\overline{B_{in}(all)}} \right) \quad (31)$$

The $\overline{B_{in}(sample)}$ is the estimation of the average upload bandwidth through l sample measurements (peers), while $\overline{B_{in}(all)}$ is the real average of B_{in} .

Graph 2 demonstrates this estimation as a function of the size of the set of the sampled peers. As it could be observed from this graph, even in the worst case scenario, the relative error stays below 10% even if the number of sampled peers is very low and equal to 10. Moreover, the relative error further decreases to 5%, if we increase the size of sampled peers to 50. Finally, this error decreases to 2% if the number of sampled peers is set to 100. We highlight here that the

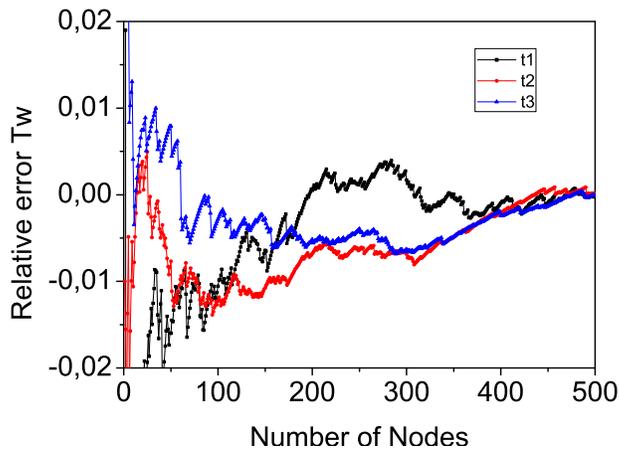
theoretical proof of the accuracy independence of our estimation algorithm from the size of the participating peers is also confirmed through our simulations.



Graph 2. Relative error of B_{in} as a function of the size of the sample

This fact testifies the outstanding scalability properties of our monitoring algorithm. It can be accurate to content distribution systems with thousands of participating peers.

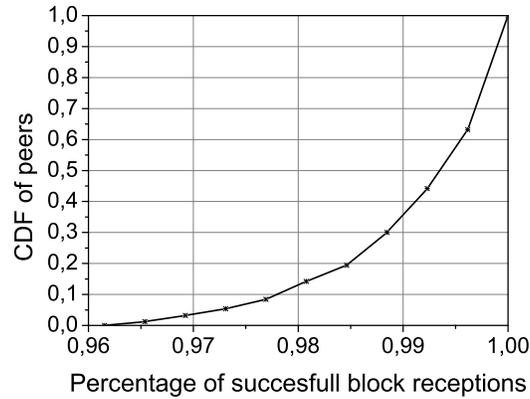
Graph 3 depicts the relative error of the estimation of T_{busy} , over different set sizes of sampled peers. In this graph are presented the values of the relative error during the three different time instances in which the convergence of the relative error was the slowest. As it can be observed from the graph, with a set size of just 50 peers, we can have an accurate estimation of T_{busy} with a relative error of less than 2%.



Graph 3. Relative error of T_w as a function of the set size of the sample

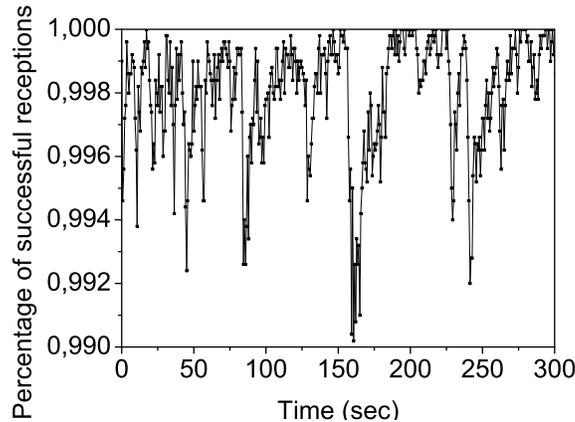
Graph 4 depicts the performance of the system. The sample size is 50 peers with a sampling period of 1 second. In this graph is presented the Cumulative Distribution Function (CDF) of the successful block receptions among the participating peers. As could be observed the mean value is more than 99%. This fact holds even in the worst case scenario, where a peer which experienced the worst performance, succeeded in acquiring more than 96% of the video blocks. This obstacle could be alleviated, meaning that every peer would have acquired the whole stream, by providing only 2% more bandwidth through the auxiliary peers. We choose to demonstrate

here a scenario where the auxiliary peers allocate bandwidth according to the aforementioned algorithms (without a robustness factor) in order to demonstrate their accuracy in the estimation of the required resources.



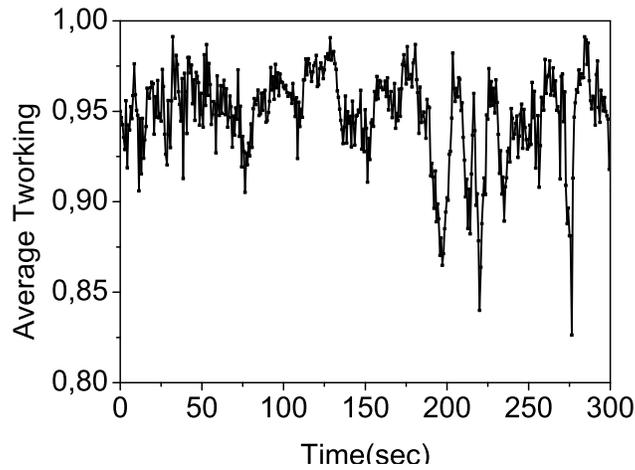
Graph 4. Cumulative Distribution Function (CDF) of successful block receptions among peers

In Graph 5 is demonstrated the percentage of the peers that received each block during the executed scenario. As it could be observed, more than 99% of the peers manage to receive each video block. This fact testifies the stability of our proposed algorithms and the uninterrupted operation of our system (independently of the average upload capacity of the participating peers).



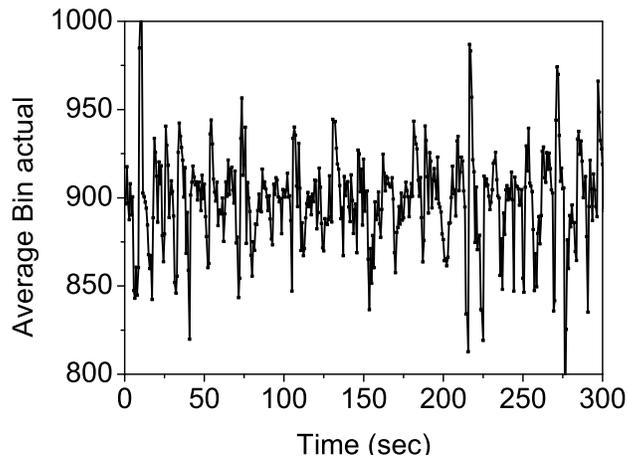
Graph 5. Percentage of the peers that received each block during the executed scenario

Our proposed algorithms rely on the provision of excess upload bandwidth from the auxiliary peers only when the aggregate bandwidth is insufficient to sustain the video distribution. The motivation behind this concept is to take full advantage of the participating peer resources. For this reason, in Graph 6 is depicted the average $T_{working}$ of the participating peers, over time. As we observe the value of this variable is with high probability more than 95%, which means that the bandwidth of the auxiliary peers is minimized while we maintain an uninterrupted streaming service.



Graph 6. Average percentage of peer's upload bandwidth utilization

In Graph 7 is demonstrated the average $B_{in,actual}$ of the participating peers over time. As could be observed, the $B_{in,actual}$ remains stable and around 900 kbps, which is the value of the video playback rate. This testifies the ability of our proposed system to offer the amount of bandwidth that is required for the efficient, continuous and complete video distribution.

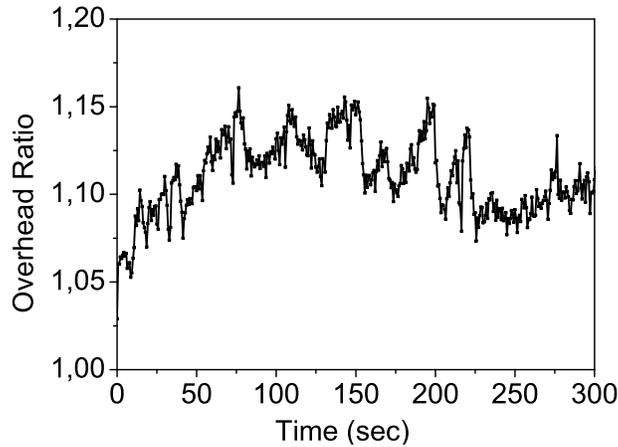


Graph 7. Average $B_{in,actual}$ during scenario's execution

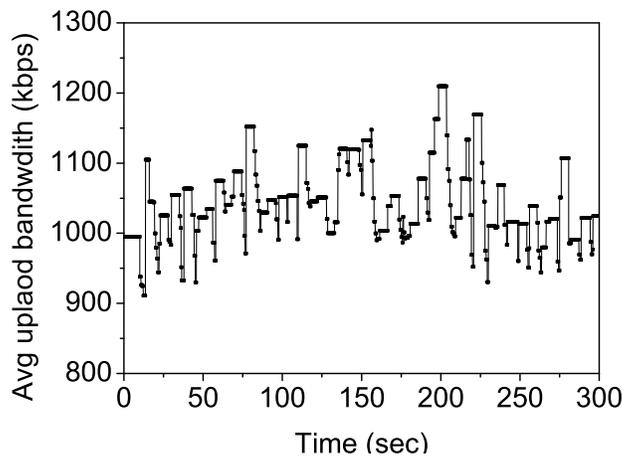
In Graph 8 is depicted the ratio between B_{in} and $B_{in,actual}$, as it is measured through the scenario's execution. This is equal to the overhead that is introduced by the scheduler (BESA) of the system. This overhead is the bandwidth needed for the transmission of the control messages and the bandwidth wasted in the transmission of duplicate packets. The purpose of this graph is to show how this ratio varies over time according to the bandwidth fluctuations.

Finally, in Graph 9 is shown the average system's upload bandwidth over time by taking into account the upload bandwidth provided from the auxiliary peers as it is calculated through our proposed algorithms. We can observe that the average is always above the rate of the video stream which is delivered. Thus, we have an uninterrupted live streaming service. We can also

observe that in time intervals where the upload bandwidth of the participating peers changes our algorithms react fast and manage to keep the average bandwidth above the desired levels.



Graph 8. Average Bin/ Bin,actual during scenario's execution



Graph 9. Average system's upload bandwidth during scenario's execution

Finally by considering that our proposed monitoring algorithm: i) is executed with a period $T=7$ seconds and ii) it probes 50 peers in order to have high accuracy the overhead will be around 7 UDP packets per second and irrelevant with the number of participating peers.

8. CONCLUSIONS

In this paper we demonstrate that exploiting scheduling fairness in a p2p live streaming system enables the estimation of the performance of such a system with low overhead and high accuracy. This makes a fair scheduler a prerequisite for future P2P live streaming systems.

Furthermore, is presented and validated an algorithm that enables the calculation of the discrepancy between the existing aggregate system's bandwidth and the required uploading bandwidth in order to sustain an uninterrupted streaming service (complete and on time video distribution to every participating peer).

This algorithm is combined with an architecture, where in case that there is a need for a supplement of upload bandwidth, exploits a set of auxiliary peers that provide dynamically this

exact amount that is required. Moreover, it is testified that our upload bandwidth allocation algorithm manages to distribute the upload bandwidth to the auxiliary peers according to their load.

Finally, by taking advantage of the properties of our proposed overlay, the system is able to ensure the efficient and optimal upload bandwidth utilization that provided by the auxiliary peers. As future work we aim to focus on time multiplexing of participating peers upload bandwidth between different video distributions. Our goal is to exploit their potential idle upload bandwidth in various video distributions in cases that the average upload bandwidth of the participating peers, in an existing video distribution, exceeds the bit rate of the video stream.

REFERENCES

- [1] VITAL++, Embedding P2P Technology in Next Generation Networks: A New Communication Paradigm & Experimentation Infrastructure, Online: <http://www.ict-vitalpp.upatras.gr>
- [2] N. Efthymiopoulos, A. Christakidis, S. Denazis, O. Koufopavlou, Liquidstream – Network dependent dynamic P2P live streaming, Springer Peer-to-Peer Networking and Applications. 4 (2011) 50-62.
- [3] R. Kumar, Y. Liu, K. W. Ross, Stochastic Fluid Theory for P2P Streaming Systems, In Proceedings of the IEEE 26th International Conference on Computer Communications, IEEE INFOCOM, Anchorage, AK, 2007, 919-927.
- [4] N. Magharei, R. Rejaie, PRIME: peer-to-peer receiver-driven mesh-based streaming, IEEE/ACM Transactions on Networking. 17 (2009) 1052-1065.
- [5] Dimitri P. Bertsekas, Network Optimization: Continuous and Discrete Models, Athena Scientific Belmont, Massachusetts, USA, 1998.
Opnet modeler, [Online]. Available: www.opnet.com
http://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test
- [6] M. Zhang, Q. Zhang, L. Sun, S. Yang, Understanding the Power of Pull-Based Streaming Protocol: Can We Do Better?, IEEE Journal on Selected Areas in Communications, 25 (2007) 1678-1694.
- [7] A. R. Bharambe, C. Herley, V. N. Padmanabhan, Analyzing and Improving a BitTorrent Network Performance Mechanisms, In Proceedings of the IEEE 25th International Conference on Computer Communications, IEEE INFOCOM, Barcelona, Spain, 2006, pp. 1-12.
- [8] F. Picconi, L. Massoulie, Is there a future for mesh-based live video streaming?, In Proceedings of the 8th International Conference on Peer-to-Peer Computing, Aachen, 2008, pp. 289-298.
- [9] A. P. C. d. Silva, E. Leonardi, M. Mellia, M. Meo, A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems, In Proceedings of the 8th International Conference on Peer-to-Peer Computing, Aachen, 2008, pp. 279-288.
- [10] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J. E. Mång, B. Melander, M. Björkman, Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering, In Proceedings of the 2006 IEEE/IFIP Network Operations and Management Symposium NOMS, 2006, Vancouver, BC, 2006, pp. 73-84.
- [11] D. Wu, Y. Liu, K.W. Ross, Queuing Network Models for Multi-Channel Live Streaming Systems, In Proceedings of the IEEE INFOCOM, 2009, Rio de Janeiro, 2009, pp. 73-81.
- [12] D. C. Tomozei, L. Massoulie, Flow Control for Cost-Efficient Peer-to-Peer Streaming, In Proceedings of the IEEE INFOCOM, 2010, San Diego, CA, pp. 1-9.
- [13] Dimitri Bertsekas, Linear Network Optimization: Algorithms and Codes, MIT Press, USA, 1991.
- [14] PPLive, [Online]. Available: <http://www.pplive.com>
- [15] PPStream, [Online]. Available: <http://www.ppstream.com>
- [16] SopCast, [Online]. Available: <http://www.sopcast.org>
- [17] TVants, [Online]. Available: <http://tvants.en.softonic.com>
- [18] A. Christakidis, N. Efthymiopoulos, J. Fiedler, S. Dempsey, K. Koutsopoulos, S. Denazis, S. Tombros, S. Garvey, O. Koufopavlou, VITAL++ a new communication paradigm: embedding P2P technology in next generation networks, IEEE Communications Magazine. 49 (2011) 84-91.
- [19] 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
- [20] P Michiardi, D Carra, F Albanese, A Bestavros (2012) Peer-assisted content distribution on a budget. ELSEVIER Computer Networks 56(7):2038–2048

- [21] H Kavalionak ,AMontresor (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
- [22] D Wu, Y Liang, J He, X Hei (2013) Balancing Performance and Fairness in P2P Live Video Systems.IEEE Circuit and Systems for Video Technology 23(6):1029 - 1039
- [23] 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
- [24] 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
- [25] Efthymiopoulos N., Christakidis A., Efthymiopoulou M., Corazza L, Denazis S., Koufopavlou O. (2015) “Congestion control for P2P live streaming”, International Journal of Peer to Peer Networks (IJP2P) 6(2) 1-21.

AUTHORS

NikolaosEfthymiopoulos received the diploma and Doctor of Philosophy degree in Electrical and Computer Engineering from the University of Patras, Greece, in 2004 and 2010, respectively. His main research interests are: network optimization, network control, scalable systems, peer to peer, distributed video streaming, distributed searching and achieving QoS in computer networks. He has more than 25 publications in these areas. He more than 10 years of experience in several FP7 ICT projects and he was technical manager assistant and WP leader in VITAL++ and STEER. He has temporarily worked as an Assistant Professor in Informatics & MM Department in Greece. He is currently technical manager in HORIZON2020’s SOCIALENERGY project and Senior Researcher at the National Technical University of Athens in Greece



Maria Efthymiopoulou received the first degree and the Doctor of Philosophy degree in Electrical and Computer Engineering from the University of Patras, Greece, in 2008 and 2015, respectively. Her main research interests are: network control, scalable systems, peer to peer, live streaming, video on demand, development of simulation environments, QoS in computer networks. She has more than 10 publications in these areas. She has 9 years of experience in several FP7 and HORIZON2020 ICT projects. She is currently a temporal assistant professor in Tei of Western Greece, Antirio, Greece.



AthanasiosChristakidis received his diploma in 2004 and his Doctor of Philosophy in 2010 from the Department of Electrical and Computer Engineering at the University of Patras, Greece. His research interests are peer to peer networks, distributed optimization, network resource allocation, and congestion control. Since 2004, he participated in several FP7 projects, and he has more than 20 publications in these areas. He has led the development of a client for stable and efficient peer to peer live streaming.

