Journal of Network and Computer Applications 45 (2014) 1–14

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Efficient content delivery scheme for layered video streaming in large-scale networks

Abbas Bradai [a], Toufik Ahmed [a], Raouf Boutaba [b,*], Reaz Ahmed [b]

[a] CNRS-LaBRI, University of Bordeaux, 351, Cours de la Libération, Talence Cedex, France
[b] School of Computer Science, University of Waterloo, 200 University Avenue W., Waterloo, ON, Canada

ARTICLE INFO

Article history:
Received 29 January 2014
Received in revised form
2 May 2014
Accepted 14 July 2014
Available online 22 July 2014

Keywords:
Content distribution
Bandwidth allocation
Scheduling
P2P
Layered video
QoS

ABSTRACT

Providing efficient layered video streaming to heterogeneous users in varying network conditions requires dynamic bandwidth allocation, efficient data scheduling and incentives. In layered streaming, the video stream is composed of hierarchically encoded sub-stream layers namely the base layer and enhancements layers. We consider a scenario where receiver peers use a pull-based approach to adjust the video quality level to their terminal and network capacities by subscribing to a different number of layers. In this context, in order to take advantage of the available bandwidth in the network and to enhance end users Quality-of-Experience (QoE), we propose a novel approach that efficiently allocates sender peers' upload bandwidth to receiver peers. The upstream peer bandwidth is allocated depending on the quality level (requested layers) of the receiver peers, starting by allocating bandwidth for the lower layers first. In order to allocate bandwidth for a certain layer, an auction game is established to distribute the bandwidth among the receiver peers, where the sender peers “sell” their items (upload bandwidth) according to bids submitted by receiver peers. The main goal of this approach is to favor high priority peers while ensuring a minimum quality level to all peers. The proposed bandwidth allocation mechanism is paired with efficient scheduling mechanism for layered streaming. It aims to fully take advantage of the allocated bandwidth while respecting the layers dependency of the stream and the data blocks playback deadline. Extensive evaluations are conducted to compare our proposed algorithm with other bandwidth allocation strategies for layered video streaming. The obtained results show the effectiveness of our model in terms of video quality, useless chunks ratio and bandwidth utilization under different network/streaming conditions.

1. Introduction

Recent years witnessed a spectacular increase in demand for customized video streaming services. For a large part, this demand is characterized by the wide availability of wireless access networks and the proliferation of various mobile devices (e.g., smart phones, tablets, gaming consoles, etc.) with different CPU, memory, and network capacities.

Layered video streaming, such as Scalable Video Coding (SVC) (Cheng-Hsin and Hefeeda, 2008), provides a convenient way to perform video quality adaptation to adjust to the end device heterogeneity and changing network conditions. Layered video streaming consists of a base layer and multiple enhancement layers. Receivers can adjust the video quality level to their capability by subscribing to a different number of layers.

Besides, using peer-to-peer overlay has become more and more popular approach for streaming video content over the Internet due to its high scalability and facility of deployment. In P2P streaming, peers actively contribute their resources (mainly upload bandwidth) by forwarding their available content to their connected peers. Since the cumulative available resources in this approach grow with the user population, this approach can scale with the number of joining peers in the session.

Thus, streaming a layered video over P2P architecture is promising approach for scalable video delivery to a large number of heterogeneous receivers (Guo et al., 2009; Liu et al., 2009; Ramzan et al., 2011). An example of such architecture is provided in Fig. 1. In this example the video source streams a layered video composed of three layers. The peers subscribe to number of layers depending on their capacities and each receiver peer becomes in its turn sender peer by serving one or many layers.

1.1. Motivations

In P2P video streaming systems, the content retrieval mechanism allows a user to receive streaming data blocks (chunks) from

* Correspondence to: Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1. Tel.: +1 519 888 4820.
E-mail addresses: bradai@labri.fr (A. Bradai), tad@labri.fr (T. Ahmed), rboutaba@uwaterloo.ca (R. Boutaba), r5ahmed@uwaterloo.ca (R. Ahmed).

http://dx.doi.org/10.1016/j.jnca.2014.07.004
1084-8045/

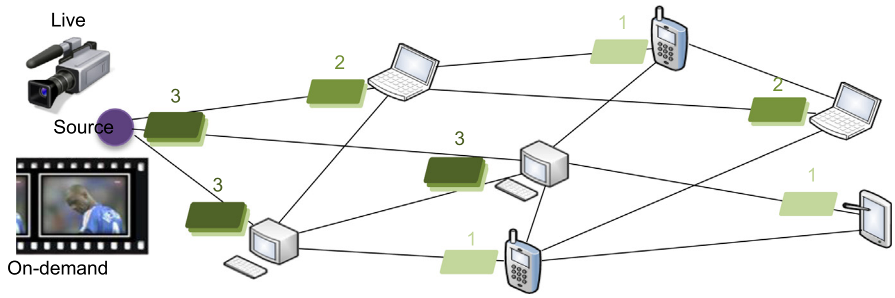

Fig. 1. Example of P2P layered streaming architecture.

other peers using the constructed overlay. This mechanism plays a leading role in the video streaming process and its efficiency influences the global performance.

Efficient chunks scheduling and appropriate bandwidth allocation are the two significant challenges of the content retrieval mechanism for real-time P2P streaming systems. The two components cannot be dissociated each from the other. Indeed, a good exploitation of the sender peers' bandwidth cannot be reached without optimized scheduling scheme. Reciprocally, an efficient scheduling algorithm cannot achieve a good throughput without appropriate sender peers' bandwidth allocation mechanism.

While considering the dynamicity of peers in terms of joining/leaving the network, the heterogeneity of peers and their need for bandwidth, an efficient and dynamic mechanism for bandwidth allocation is required to ensure the timely availability of the streaming content and acceptable quality level for peers while fully taking advantage of the overall bandwidth in the network.

In the context of pull-based layered streaming over P2P architecture, a peer commonly requests video layers from different upstream peers, and each upstream peer shares its upload bandwidth among different downstream peers serving different layers. Consequently, resolving bandwidth conflicts among peers in order to maximize the benefits of both upstream and downstream peers is a very challenging problem, because of the layers importance, their dependencies and the peers' priorities.

On the other hand, the scheduling task is complicated in the context of layered video streaming since chunks received after their playback deadline are not played and considered as useless chunks. In addition, chunks of higher layers received without their corresponding chunks from higher layers are not played also and considered as useless too.

Moreover, selfishness of peers in P2P networks is inevitable (Park and van der Schaar, 2010). Therefore, a key research question to consider when designing streaming system in P2P architecture is: “How to exploit and manage the selfishness of peers in order to reduce the global streaming cost while satisfying the peers quality level requirements and priorities?”

Most related works considered one or two of the three components of the content delivery mechanism of the P2P layered streaming systems: bandwidth allocation, the chunks scheduling or the incentives. But no one considered the three issues all together, which is particularly challenging for efficient P2P streaming. Hence, the main challenge in this work is how to allocate the upstream bandwidth among its downstream peers (see Fig. 2) while fully taking advantage of the available bandwidth in the network and handling the peers selfishness to ensure the better video quality for all peers.

In our previous work (Bradai and Ahmed, 2012a), we presented an efficient chunks scheduling mechanism for P2P layered streaming systems. In this work, we mainly focus on the bandwidth allocation and the incentive mechanisms.

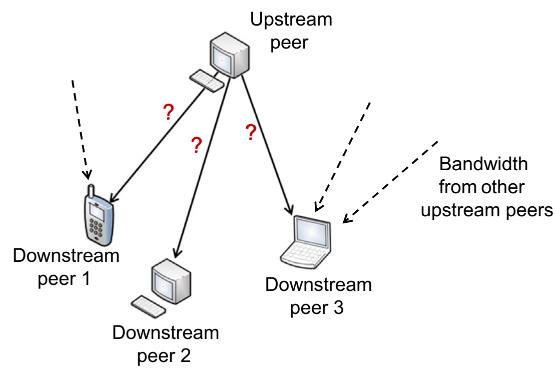

Fig. 2. Example of upstream peer bandwidth allocation problem.

1.2. Contributions

Recently there have been significant research efforts on designing P2P architectures by addressing the selfishness of peers. They can be categorized into two main approaches: “non-strategic behavior approach” such as Ma et al. (2004), Wang and Li (2005), Cui et al. (2006) and “strategic behavior approach” such as Takahashi and Tanaka (2001), Chun et al. (2005). In the former, each peer is considered as a potential game player aiming at maximizing its utility regardless of the behavior of other peers, while in the later each peer aims at maximizing its utility taking into consideration the actions of the other peers. In this paper, we propose a novel strategic and incentive model for efficient bandwidth allocation in P2P layered streaming networks.

Our main contributions in this paper are as follows:

(1) Auction game model for bandwidth allocation in P2P layered streaming is proposed. The players in this game are the upstream peer which sell its upload bandwidth and on the hand the downstream peers which bids for. The proposed model is content aware and an auction game is set up for each layer.
(2) Water filling algorithm based downstream peer bidding strategy is proposed. It aims to request the stream from the best links in terms of QoS metrics such as packet loss ratio and end to end delay.
(3) A theoretical study of the proposed mechanism is provided and the Nash equilibrium of the system is proved.
(4) Extensive simulations are conducted to show the performance enhancement of our mechanism using SVC video sequences and real P2P streaming system traces.

This paper extends our previous work (Bradai and Ahmed, 2012b) in many ways. It provides a more detailed description of the problem at hand and our solution approach including a mathematical analysis. In this regard, we provide a detailed proof of Nash equilibrium of our system. It also presents a new incentive mechanism to motivate peers to share their bandwidth. Moreover, new extensive simulations are performed to evaluate the effectiveness of our proposal under different network and streaming conditions, comparing with state of the art and other proposed strategies, using real world streaming systems traces.

1.3. Paper organization

The rest of the paper is organized as follows. Section 2 introduces the background and discusses related work. Section 3 presents the system model for layered P2P streaming and problem formulation. Section 4 describes our proposed bandwidth allocation and discusses its Nash equilibrium. Section 5 presents briefly the scheduling mechanism. Section 6 presents our comparative evaluations and results. Finally, Section 7 concludes this paper.

2. Related work

In the literature, content delivery mechanisms in P2P streaming (Karunakar et al., 2012) systems can be classified to three main categories: (a) bandwidth allocation mechanisms; (b) chunks scheduling mechanisms; and (c) incentive mechanisms.

A number of studies investigated different bandwidth allocation strategies and their impact on network performance and on end users satisfaction.

Bradai and Ahmed (2012b), Chun et al. (2005) and Manzillo et al. (2012) studied the impact of peers' upload bandwidth and established conditions for universal streaming for churnless systems. Furthermore, measurement studies and implementations (Hei et al., 2007; Li et al., 2008) confirmed that bandwidth has a big impact on streaming quality for P2P streaming systems. Li et al. (2008) studied the Coolstreaming system by exposing its design rationale. They showed that there is a highly distorted resource distribution in such systems and the performance is mostly affected by the system dynamics.

Recently, researchers have studied bandwidth allocation for improving streaming quality in more challenging P2P networks such as multi-overlay, multi-sources and multi-swarm P2P streaming systems. Wu et al. (2008) studied the bandwidth contest among coexisting overlays and proposed an auction-based solution. Liang et al. (2011) studied optimal bandwidth sharing in multiple video conferencing swarms systems. They dynamically share a pool of helpers between swarms to address the intra and inter-swarms bandwidth shortage. However, none of these works have taken into consideration the properties of layered video streaming, and its potential benefits for providing personalized and customized video quality while coping with the bandwidth fluctuation problem.

For chunks scheduling, many valuable mechanisms have been proposed (Zhou et al., 2009; Lee, 2010; Zhang et al., 2009; Szkaliczki et al., 2010; Zhang et al., 2012). Zhou et al. (2009) propose an optimal scheduling strategy to minimize the overall video distortion, but the approach is strongly related to the Multiple Description (MD) coding, which is less efficient compared with layered coding (Lee, 2010). Zhang et al. (2009) have discussed the scheduling problem in data-driven streaming systems. They define a utility for each chunk as a function of its rarity, which is the number of potential senders of this chunk, and its urgency, which is the time difference between the current time and the deadline of this chunk. They then use this model to transform the chunk scheduling problem into a min-cost flow problem. This algorithm, however, is computationally expensive and may not be feasible for live video streaming systems subject to strict deadlines on computationally-constrained devices. Szkaliczki et al. (2010) address the chunk selection problem in streaming layered video content over peer-to-peer networks. The authors present a number of theoretical solutions to maximize the utility function of chunks that exist in the literature. Zhang et al. (2012) propose SVC content segmentation algorithm to overcome the problem of equal playback length segmentation. The proposed mechanism splits each layer into chunks with equal size and tags each chunk with a time-stamp according to its playback deadline. Based on these time stamps, they propose a scheduling algorithm and organize the buffer into two stages: Elementary Layers and Excess Layers in order to adapt to the dynamic network conditions. However, authors do not explain how the upload peer's bandwidth is allocated.

To handle the free riding phenomenon (Ma et al., 2006) in P2P networks, two categories of incentive mechanisms have been considered in the state of the art: rating and monetary payment (Park and Schaar, 2010). In the rating category, each peer is rated regarding its upload history or downloads actions. Based on that, the supplying peer decides the amount of allocated bandwidth for the requester peer (Habib and Chuang, 2004; Iosifidis and Koutsopoulos, 2008; Liang et al., 2010; Zhang et al., 2005; Zhao et al., 2012). This approach presents mainly two drawbacks. Firstly, some malicious peers could have more allocated bandwidth from the network by providing false ratings for other peers (Rückert et al., 2012). Secondly, it penalizes peers of low bandwidth capacity in the network (Park and Schaar, 2010). In the monetary payment category (Qiu and Srikant, 2004), the peers pay for their download and are rewarded for their upload. A virtual currency or even real currency could be used for payment. In this work, we adopt a credit-based incentive mechanism to incite peers collaboration. It is a virtual currency payment incentive mechanism, in which the peers are paid for sharing their upload bandwidth. In their tour, they use the earned credit to buy their need in download bandwidth.

To our knowledge, the most closely related work to ours is the one of Wu et al. (2008). They coordinate multiple streams as an auction game, where each peer participates in media distribution by bidding for and selling bandwidth. Their strategy is not appropriate for scalable streams as each stream is considered as an isolated stream without any relationship with other streams. We argue that there exists inherent content priority among layers. In addition, in WU approach the receiver side is not considered, to request content a simple scheduling algorithm is considered. Moreover, they only consider a scenario where the total upload bandwidth in the network is always sufficient to support all peers' requirements in all the overlays.

We note that our work does not focus particularly on the incentive mechanism, but on a combined design of bandwidth allocation, chunk scheduling with incentives in P2P layered streaming. In the following section we present the model of our system and we detail the problem of bandwidth allocation and scheduling for P2P layered streaming.

3. System model and problem statement

In this section, we first present an overview of the proposed delivery system model and considered assumptions, and then formulate the problem we would like to solve of layered streaming in P2P layered streaming systems, in its different aspects: bandwidth allocation, chunks scheduling and incentives.

3.1. System model and notation

Scalable video codes, such as SVC (Cheng-Hsin and Hefeeda, 2008), allow a video to be encoded into a base layer and several enhancement layers. The scalability of SVC is three-dimensional: quality, spacial and temporal. The quality scalability determines the video Signal to Noise Ratio (SNR); the spatial scalability determines the resolution of video in terms of height and width; and the temporal scalability determines the frame rate of the video. In our model we assume that the layered stream is encoded into M layers at the source peer $\{l_0, l_1, \ldots, l_{M-1}\}$, with l_0 representing the base layer and $l_1 \ldots l_{M-1}$ representing the enhancement layers such that l_1 is the first enhancement layer, l_2 the second enhancement layer, etc.

We assume that each video layer l_i is distributed with a transmission rate of b_i. Thus, each peer can subscribe to a particular video layer depending on its download capacity and other parameters such as its processing capability, preferences, etc. The decided quality should be dynamically adjusted to the available upload bandwidth in the receiver side. If the available upload bandwidth is not known a priori, the receiver peer should ask for the maximum quality. The chunks are then requested and delivered according to their priority in terms of layer importance and playback deadline. This will give the possibility to receive a suitable quality according to the available bandwidth. This issue was discussed in detail in our previous work (Bradai et al., 2012). Of course, contextual information such as users' preferences and terminal capabilities can also be used to determine the requested quality level.

We consider an overlay network composed of n peers relayed by a set of application-layer links $k_{i,j}$ (link between downstream peer i and its upstream peer j). So, the topology of the overlay can be modeled as a directed weighted graph $G=(S, L, K)$ where S denotes the set of upstream peers (called Seeders in some architecture), L the set of downstream peers (typically referred to as Lechers) and K the set of links.

To represent a practical network setting, we limit the upload and download capacity of any peer i by u_i and d_i respectively. Therefore, each peer can only provide limited service for its downstream peers, and make a limited layer subscription as well.

For the reader's convenience, Table 1 summarizes the notations used throughout this paper.

In the following sections, we present the bandwidth allocation problem and the chunks scheduling problem in the context of P2P layered streaming.

3.2. Bandwidth allocation problem statement

Through the following example we illustrate the problem of bandwidth allocation in P2P layered streaming. Figure 3 represents part of an overlay composed of three upstream peers A, B and C with upload capacity u_i of 100, 150 and 120 kbps respectively, and four downstream peers 1, 2, 3 and 4 with download capacity d_i of 150, 150, 100 and 100 kbps respectively.

Suppose each upstream peer allocates its upload bandwidth fairly between its direct downstream peers. In Fig. 3, the numbers on the arcs represent the distribution of the upstream upload bandwidth on their respective downstream peers by adopting this strategy. Peers 1, 2, 3 and 4 can get 100, 140, 90 and 40 kbps respectively. Suppose that the original stream is encoded into one base layer and one enhancement layer, with each layer rate at 50 kbps (i.e. $b_0=b_1=\cdots=b_M=50$ kbps). Clearly, downstream peers 1 and 2 can access the base layer and enhancement layer l_1, while downstream peer 3 can get only the base layer and downstream peer 4 cannot get any layer. In this case the unused bandwidth w is

$$w=(u_A+u_B+u_C)-(l_0*b_0+l_1*b_1+l_0*b_0+l_1*b_1+l_0*b_0)=120 \text{ kbps}$$

An optimized allocation strategy could be the one represented in Fig. 4, where downstream peers 1, 2 and 3 can get layers l_0 and l_1 and peer 4 can get the base layer l_0. The unexploited bandwidth in this case is 20 kbps instead of the 120 kbps in the first strategy.

Following the above definitions and assumptions, we now formulate the problem as follows.

Given a set of upstream peers S willing to share a total upload bandwidth $U=\sum_{i \in S} u_i$, and a set of downstream peers requesting each a bandwidth d_i corresponding to quality level Q_j (i.e., requesting layers: $l_0, l_1, \ldots, l_j$), we seek the most optimal possible solution which allocates the upload bandwidth U in such a way that takes better advantage of the available bandwidth and maximizes the total system utility in terms of downstream peers' quality level satisfaction.

Taking better advantage of the available bandwidth means reducing the amount of unused data due to either over-

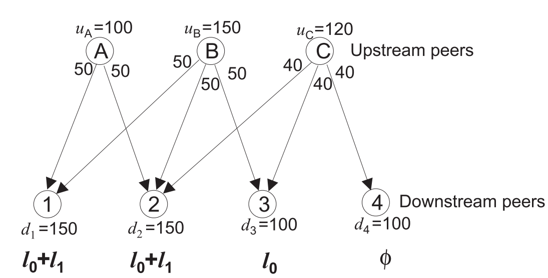

Fig. 3. Example of native bandwidth distribution in P2P layered streaming.

Table 1
Notation table.

S	Set of upstream peers
L	Set of downstream peers
K	Set of application links
l_i	Video layer i, $i \in \{0, 1, 2, \ldots, M-1\}$, the stream is encoded into M layers: $l_0, l_1, \ldots l_{M-1}$
b_i	Bit rate of layer l_i
pr_j	Priority of downstream peer j
u_i	Upload bandwidth capacity of peer i
d_i	Download bandwidth capacity of peer i
d_i^j	Bandwidth allocated to the downstream peer i by its upstream peer j
$b_{i,j}^k$	Bandwidth requested by the downstream peer i to its upstream peer j to acquire layer k
$p_{i,j}^k$	Unit price that the downstream peer i is willing to pay for the upstream peer j to acquire bandwidth for layer k
l_M^i	Maximum layer available at peer i
u_i^k	Peer i's upload bandwidth allocated to layer k
L_j	Set of downstream peers connected to upstream peer j
S_i	Set of upstream peers of the peer i
T_i	Initial budget allocated to peer i
Q_j	Quality level j, i.e., layers: $l_0, l_1, \ldots, l_j$

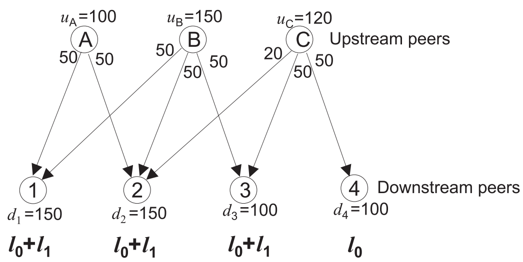

Fig. 4. Example of optimized bandwidth distribution in P2P layered streaming.

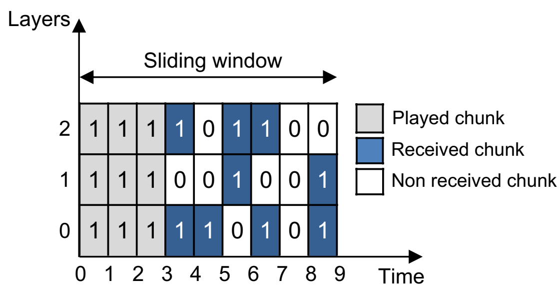

Fig. 5. Sliding window mechanism.

allocating bandwidth to peers or receiving layers without their corresponding lower layers. The system-level utility optimization can be formulated as

$$\min\left(\sum_{i \in S} u_i - \sum_{j \in L} \sum_{m=0}^{m=M^j} (l_m * b_m)\right) \tag{1}$$

Subject to

$$\sum_k u_i^k \le u_i \text{ (for all } i \in S \tag{2}$$

and

$$\sum_{m=0}^{m=M^j} (l_m * b_m) \le d_j \tag{3}$$

where M^j represents the maximum layer played by peer j, and u_i^k represents the peer i upload bandwidth allocated to the layer k.

Eq. (2) ensures that the upload bandwidth of the upstream peer i is not violated, and Eq. (3) ensures that the download capacity of a peer j is not violated.

In pull-based P2P streaming systems, it is up to the receiver peer to request video data from the sender peer. Hence, after that the upstream peers allocate its bandwidth, the downstream peer request the video chunks using a scheduling mechanism. The problem of chunk scheduling is briefly introduced in the next section. We note that the adopted scheduling mechanism was presented in our previous work (Bradai and Ahmed, 2012a).

3.3. Chunks scheduling problem statement and buffer structure

In pull-based P2P streaming systems, it is up to the downstream peer to request the video chunks, which depend on the links capacity allocated in the bandwidth allocation phase. To better explain the problem of scheduling, we assume a mesh-based pull approach in which the receiver side buffer is organized into a sliding window (Fig. 5) containing chunks of different layers. The chunks beyond the playhead position are denoted as the exchanging window; only these chunks are requested if they have not been received yet (the chunks whose deadline has passed will not be requested). Each peer periodically announces the chunks that it holds to all its neighbors by sending a buffer map (Fig. 5), a bit vector in which each bit represents the availability of a chunk in the sliding window. Periodically, each peer sends requests to its neighbors for the missed chunks in its exchanging window. As long as its request remains in the exchanging window, chunks are re-requested if not received.

Of course, upper layer chunks received without the corresponding lower layer chunks are not decodable (and are considered useless, as described earlier). Thus, the chunks having time stamp $T=5$ in Fig. 5 are not played, because the base layer was not received. In order to increase the throughput of the system, our approach aims to take full advantage of the download bandwidth of peers by maximizing the number of chunks that are requested within each scheduling period.

In the following section we present our solution for efficient layered streaming over P2P networks. We detail the bandwidth allocation mechanism with incentives and we overview the chunks scheduling mechanism presented in detail in our previous work (Bradai and Ahmed, 2012a).

4. Bandwidth allocation mechanism

We propose a sender peer bandwidth allocation for mesh pull-based architectures. In this section, we develop the sender side bandwidth allocation strategy. The sender peer allocates its bandwidth periodically (long-term allocation), and it is up to the receiver peers to request the appropriate chunks. Figure 6 summarizes the relationship between the bandwidth allocation module and the chunks scheduling module. The scheduling module aims to better request the chunks in timely and efficiently manner depending on the available bandwidth in the neighbors. The bandwidth is allocated by the sender peer using the bandwidth allocation module.

The main goal of the bandwidth allocation module is to allocate the sender peers upload bandwidth in such a way that ensures a minimum quality level to all peers and higher quality if the residual bandwidth allows, while differentiating the service quality received by the peers based on their bandwidth contribution in the network. To do so, the allocation process is performed layer by layer, starting from the lower to the higher layers. Each sender peer tries first to ensure the base layer for all of its neighbors, and then, if bandwidth remains, allocates it for the higher layers.

Bandwidth allocation for a certain layer is modeled as a set of dynamic auctions organized by the upstream peers in order to give rise to competition for its upload bandwidth u_j.

In this work we adopt the *English auction* (Coppinger et al., 1980), where the auctioneer opens the auction by announcing an initial price and then accepts progressively higher bids from interested buyers. The players in these auctions are the downstream peers. Indeed, each downstream peer, having an initial budget T_i, submits bids to all its downstream peers in order to purchase bandwidth. Consequently, each downstream peer can participate in different independent parallel auctions A_j. The upstream peers aim to maximize their profit, and allocate their upload bandwidth for the higher bidder downstream peer.

We note that the proposed bandwidth allocation mechanism is executed periodically in order to deal with the bandwidth fluctuation. In addition, the mechanism is triggered by an upstream peer in the case of new peer joining the system in order to allocate a bandwidth to the new arriving peer. Similarly, in the case of peer living the system (peers churn), the mechanism is called in order to re-allocate the bandwidth consumed by this peer to the remaining downstream peers. Thus, the proposed mechanism allows to regulate the bandwidth allocation and deal with the extreme situations, namely the flash crowd phenomena where

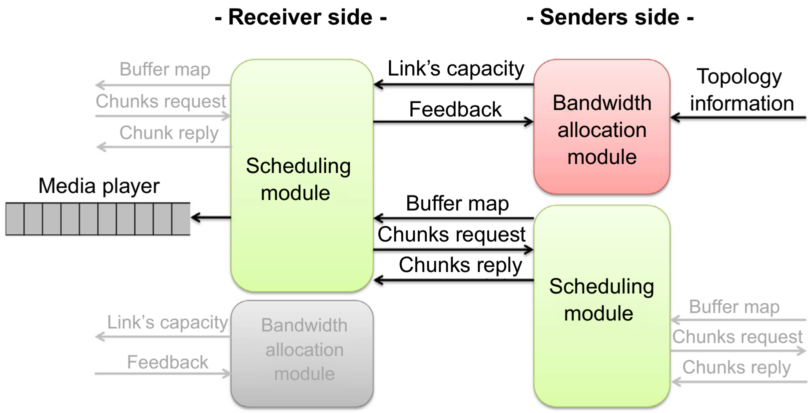

Fig. 6. Bandwidth allocation and chunks scheduling modules interaction.

peers join the system massively and suddenly, and leave it in the same manner.

In the following sections, we detail the bandwidth allocation process. We start by giving an overview of the proposed mechanism then we detail the per-layer allocation process.

4.1. Credit-based incentive mechanism

To avoid selfishness or free-rider, we introduce a credit-based incentive mechanism. It is on the common sense that the peer using the bandwidth should be charged and the peer providing the bandwidth should be rewarded. Credit represents a kind of virtual currency that can only be used in P2P networks for payment. If the peer requests a video layer from the neighbor, it has to pay for it in credits. For example, when player j receives its bandwidth assignment x_{ij}^m from peer i, it pays an amount $P_{ij}^m \times x_{ij}^m$ of credits to peer i. correspondingly, the credits of peer i is increased with equivalent amount. In this way, the peer is motivated to share the bandwidth to earn credits. In the following section we present the bandwidth allocation mechanism along with incentives.

4.2. Bandwidth allocation: a content-aware approach

In order to guarantee a minimum quality level for all downstream peers and respect the layers dependency, every upstream peer should start by allocating bandwidth for the base layer, then for the enhancement layers in an ascending order. In our model, every upstream peer organizes an auction to distribute bandwidth needed for the base layer. Then, if there is still remaining bandwidth, it organizes another auction to sell bandwidth for the first enhancement layer, the second enhancement layer, and so on. In the receivers' side, the peers participate in auctions depending on the quality level they need. A peer requiring a quality level 2 will participate in a set of auctions organized by its upstream peers to distribute bandwidth for base layer (l_0), for the first enhancement layer (l_1) and for the second enhancement layer (l_2). We note that an upstream peer does not start allocating bandwidth for an upper layer, until there is no request for the current layer from their downstream peers (i.e., until the downstream peers' requests are satisfied or their budget is exhausted).

The auction game to distribute bandwidth for certain layer l_i is illustrated in Fig. 7. In this figure, three parallel auctions are organized by upstream peers A, B and C in order to allocate their upload bandwidths u_A, u_B and u_C respectively for layer l_i. The downstream peer 1, connected to upstream peers A, B and C, participates in auctions A_1, A_2 and A_3 organized by the three upstream peers. While the downstream peer 4 participates only in auction A_3 organized by C (since peer 4 is connected only to C).

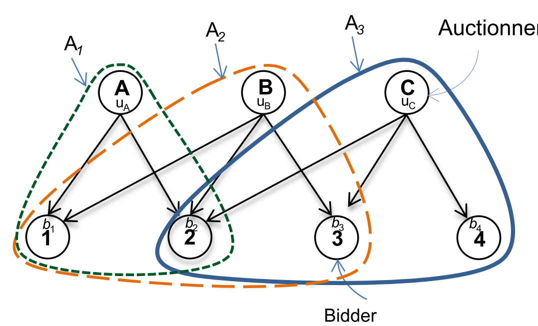

Fig. 7. Auctions example.

Let $b_{i,j}^k$ be the bandwidth requested by the downstream peer i to its upstream peer j to acquire bandwidth for layer k and $p_{i,j}^k$ be the unit price that the downstream peer is willing to pay for that bandwidth. The bid of the downstream peer i can be expressed by the tuple $B_{i,j}^k = (b_{i,j}^k,\ p_{i,j}^k)$.

In this section we modeled the bandwidth allocation problem in P2P layered streaming as a set of auction games, where items to sell are the upload bandwidths, and where the sellers are the upstream peers and the buyers are the downstream peers. We discuss in the following the allocation strategies of the upstream peers and the bidding strategies followed by the downstream peers.

4.2.1. Upstream peers' bandwidth allocation strategy

An upstream peer starts allocating bandwidth primarily to the lower layers then to the upper layers in ascending order. It executes the algorithm presented in Algorithm 1.

Algorithm 1. Allocation strategy-global algorithm

$l_k = l_0$
While $\boldsymbol{u_i} > 0$ and $\boldsymbol{l_k} \le \boldsymbol{l_M^i}$
  Auction (l_k)
 $u_i = u_i - u_i^k$
 $l_k = l_{k+1}$
End while

where u_i^k represents the total bandwidth allocated for layer k to all downstream peers of peer i and l_M^i is the maximum layer available in peer i.

In the following, we detail the strategy of the upstream peer within the auction game to allocate bandwidth for a layer k.

In auction A_j^k, organized by peer j to allocate bandwidth for layer k, the seller j aims to maximize its revenue by selling its bandwidth at the best price. Given the downstream peers' bids $B_{i,j}^k = (b_{i,j}^k,\ p_{i,j}^k)$, the upstream peer j aims to maximize

$$\max \sum_{i \in L_j} (b_{i,j}^k * p_{i,j}^k) \tag{4}$$

Subject to

$$\sum_{i \in L_j} b_{i,j}^k \le u_j \tag{5}$$

where L_j denotes the set of downstream peers connected to the upstream peer j.

In order to maximize its revenue, the upstream peer adopts the best offer auction strategy: it starts first by serving the downstream peer, willing to pay the highest price. Once it is served and if there is still remaining bandwidth, the downstream peer proposing the second highest price will be served and so on.

The allocation strategy of the bandwidth for a layer k is performed in many rounds as shown in Algorithm 2.

Algorithm 2. Per layer bandwidth allocation strategy

(1) Receive bids from downstream peers
(2) Allocate bandwidth to the downstream peer willing to pay the highest price
(3) While there is still remaining bandwidth serve the downstream peer willing to pay the next highest price
(4) Notify the allocated bandwidth to all the downstream peers involved in the auction
(5) Receive new bids from downstream peers (having sufficient budget) whose bandwidth request is not satisfied. Go to 2

4.2.2. Downstream peers' bidding strategy

A critical question is how much bandwidth a down-stream peer requests to each of its upstream peer $b_{i,j}^k$ and at what price $p_{i,j}^k$. The ultimate goal of the downstream peer is to minimize the bidding cost and thereby the streaming cost. The bidding price can be mapped as in a real-world auction to the items' purchase prices which reflect the competition degree on these items. Indeed, we believe that requesting a stream from the less loaded upstream peers, allows to reduce the delay, since it reduces peer congestion by balancing the load over the concerned peers. In addition, load balancing in the streaming overlay allows to minimize the impact of peer churn.

The streaming cost can be modeled as the transport cost of the purchased items. In the context of P2P streaming, reducing the streaming cost is equivalent to receiving the stream over the best links, that is the links with lowest delay, lowest bit error rate, etc. As such, the goal of the downstream peer is to get bandwidth from less loaded upstream peers (lowest price) and through the best links (lowest streaming cost).

In our system, the downstream peer starts first by requesting bandwidth for the lower layers, and then incrementally for the enhancement layers, by joining the corresponding auctions organized by the upstream peers in this order. This strategy allows requesting primarily bandwidth for the lower layers over the best links and then the enhancement layers over the remaining links. Hence, lower layers have more chance to be received by the downstream peers, and consequently more chance to be decoded properly. In contrast, if the upper layers are those prioritized, the decoding of the stream becomes impossible if the corresponding lower layers are missing and the throughput of the system will degraded as a consequence.

Formally, in each auction organized by an upstream peer j to allocate bandwidth for layer k, the downstream peer aims to minimize the bidding cost. That means

$$min \sum_{j \in S_i} (b_{i,j}^k * p_{i,j}^k) \tag{6}$$

Subject to

$$\sum_{j \in S_i} b_{i,j}^k \ge b_k \tag{7}$$

where S_i denotes the set of upstream peers of peer i. Condition (6) ensures that the downstream peer i gets the necessary bandwidth to play layer k.

In addition to the bidding cost, the downstream peer also aims at minimizing the streaming cost from each of its upstream peer j, denoted as $E_{i,j}(b_{i,j}^k)$. Therefore, the bidding strategy of downstream peer i, to acquire bandwidth for layer k, can be seen as an optimization of the overall cost problem:

$$min \sum_{j \in S_i} (b_{i,j}^k * p_{i,j}^k + E_{i,j}(b_{i,j}^k)) \tag{8}$$

Subject to

$$\sum_{j \in S_i} b_{i,j}^k \ge b_k \tag{9}$$

$$b_{i,j}^k \ge 0 \tag{10}$$

In practice, we consider the streaming cost function $E_{i,j}$ as a non-decreasing function depending on $b_{i,j}^k$, strictly convex and twice derivable.

In the following we present the bidding strategy of the peer to set the requested bandwidth $(b_{i,j}^k)$ and the bidding unit price $(p_{i,j}^k)$.

4.2.2.1. Peer's strategy to set the requested bandwidth $(b_{i,j}^k)$. Given the bid price p_j^k in an auction organized by the upstream peer j to allocate bandwidth for layer k, the downstream peer i aims to optimize the overall cost by adjusting the requested bandwidth $b_{i,j}^k$ from each downstream peer. Hence, the goal of the downstream peer is to minimize the global marginal cost M_i defined as the change in total cost that arises when the requested quantity changes by one unit (White, 2010).

Let c_i be the overall cost at the downstream peer i, i.e.,

$$c_i = \sum_{j \in S_i} (b_{i,j}^k * p_{i,j}^k + E_{i,j}(b_{i,j}^k)) \tag{11}$$

The corresponding marginal cost is

$$M_i = \frac{dc_i}{db_{i,j}^k} = \sum_{j \in S_i} p_{i,j}^k + \sum_{j \in S_i} \frac{dE_{i,j}(b_{i,j}^k)}{db_{i,j}^k} \tag{12}$$

Since the streaming cost function $E_{i,j}$ is strictly convex, the second derivative $dM_i/db_{i,j}^k = d'' E_{i,j}(b_{i,j}^k)/db_{i,j}^k$ is strictly positive.

Consequently, the marginal cost M_i increases with the increase of the bandwidth request $b_{i,j}^k$.

To efficiently solve this optimization problem we adopt the water-filling algorithm (Boyd, 2004). To set the bandwidth quantity (to request from an upstream peer j), the downstream peer i – applying the water-filling algorithm – sets $b_{i,j}^k$ to 0 for all $j \in S_i$, then identifies the upstream peer j_0 having the lowest marginal cost M_{i,j_0} and increases its demand b_{i,j_0}^k until the marginal cost becomes equal to the next highest marginal cost M_{i,j_1}, corresponding to the upstream peer j_1 (Fig. 8). The downstream peer i increases then fairly b_{i,j_0}^k and b_{i,j_1}^k until their corresponding marginal cost M_{i,j_0} and M_{i,j_1} meet the next highest marginal cost M_{i,j_2} corresponding to the upstream peer j_2 (Fig. 9), and so on. The downstream peer carries out this mechanism with all of its upstream peers until it obtains the bandwidth that it requests for layer k (b_k).

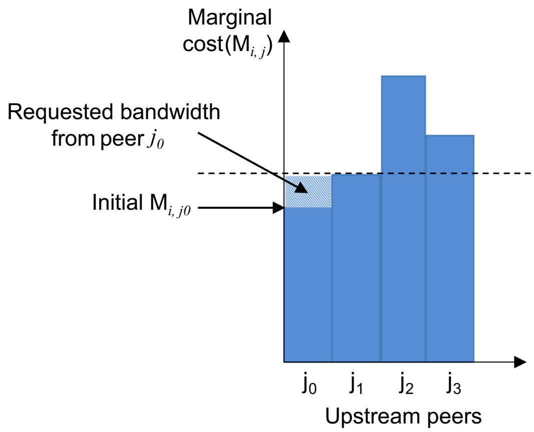

Fig. 8. Peer i water filling algorithm example (Step 1).

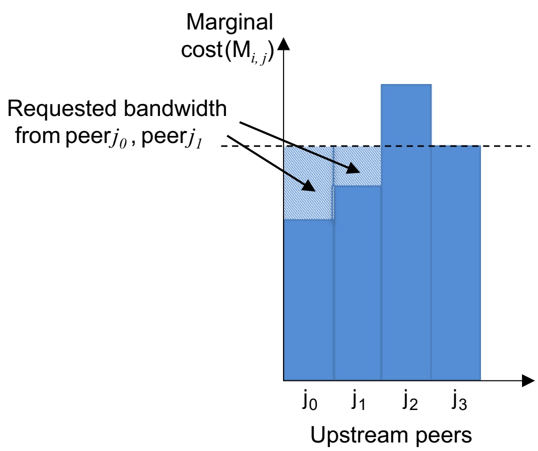

Fig. 9. Peer i water filling algorithm example (Step 2).

4.2.2.2. Downstream peers strategy for setting the bidding unit price ($P_{i,j}^k$). After defining the bandwidth request strategy of the downstream peer, the next question is how the downstream peer sets the unit price that it will announce to the upstream peer j?

In the downstream strategy (Section 4.2.2), we have seen that the upstream peer aims to maximize its benefits by selling its bandwidth at the higher price. The collected amount will constitute its budget T_i^k to buy bandwidth from other peers.

When the downstream peer i joins the auction organized by an upstream peer j, first, it sets its price bid to one unit (i.e.$p_{i,j}^k=1$). Using the water-filling algorithm described earlier, it computes the optimal quantity of bandwidth $b_{i,j}^k$ to request from each upstream peer j, and submits bids accordingly. After the upstream peers allocate their upload bandwidth using the strategy described above, it proposes the bandwidth $a_{i,j}^k$ to the corresponding downstream peers. On receiving the proposed bandwidth, the downstream peer determines its behavior in the next round of auctions, using the following algorithm:

Algorithm 3. Bidding algorithm

(1) Receive bandwidth allocation and current prices from upstream peers.
(2) For each upstream peer:
If requested bandwidth $b_{i,j}^k$ from an upstream peer j is not satisfied:
(a) Increase the price $p_{i,j}^k$ by one unit within the reference price $\tilde{p}_i^k$
(b) Using the water-filling algorithm, decide the quantity of bandwidth $b_{i,j}^k$to request from j

(3) Send new bids ($b_{i,j}^k$, $p_{i,j}^k$) to the upstream peer

It is clear in this algorithm that the reference price $\tilde{p}_i^k$ assigned to the downstream peer allows to differentiate between the downstream peers according to their priorities.

4.3. Convergence to Nash equilibrium

Our mechanism for bandwidth allocation can be modeled as a non-cooperative game where the players are the downstream peers in the set L, the strategies are the bids ($b_{i,j}^k$, $p_{i,j}^k$) and the cost function of a player i is the overall streaming and bidding cost C_i. Formally, we consider the finite game $\Gamma = <L, D, C>$ where

- L denotes the set of players (downstream peers)
- D denotes the set of strategies, i.e. $D=(D_1, D_2, \ldots, D_i)$, where $D_i=(B_{i,1}^k, B_{i,2}^k, \ldots, B_{i,j}^k)$ is the tuple of bids submitted by the player to its upstream peers.
- C denotes the set of costs: $C=(c_1, c_2, \ldots, c_i)$, where c_i is the overall cost at player i as defined in (10).

Theorem. *The formulated auction game for bandwidth allocation in layered P2P streaming leads to a Nash equilibrium.*

Proof. A detailed proof of the theorem is provided in the Appendix.

5. Chunks scheduling mechanism

The main goal of the scheduling mechanism is to efficiently request the missing chunks in the receiver peer buffer. This can be achieved by requesting the higher priority chunks before the lower priority chunks while at the same time taking full advantage of the available links capacity. Since, this scheme will closely depend on the definitions of these priorities, we now explain how they are calculated.

Intuitively, it seems clear that since chunks are useless if they are not decoded by their playback deadline, the priority of each chunk should be closely related to how close they are to it. Another issue to consider is the dependency between layers; a higher layer chunk received without its corresponding lower layer chunks will not be decoded. To factor these two variables into our priority model, we will define two functions. The first one, the *emergency priority* P_E, is a function of how close a chunk is to its playback deadline; the second one, the *layer priority* P_l, is a function of how many underlying layers are necessary to decode a particular chunk. Using these two functions, we can define our priority function P_{ij} as

$$P_{ij} = P_E(T_i - D_j^i) + \theta P_L(l_j) \quad (13)$$

where T_i denotes the current time in the peer i, D_j^i denotes the playback deadline of chunk j in peer i, l_j denotes the stream layer to which chunk j belongs, and θ is a parameter that can be adjusted for different layers prioritization strategies.

Hence, $P_E(T_i - D_j^i)$ evaluates P_E at a time interval equal to the remaining time that chunk j has until its playback deadline at peer i, and $P_l(l_j)$ evaluates P_l at an integer proportional to the number of underlying layers needed to decode chunk j.

After defining the chunks priority function, we model the chunk scheduling in layered P2P streaming systems as assignment problem. Where a set of chunks are to be assigned to a set of neighbors. The main goal is to maximize the chunks priority sum. For that purpose we define the decision variable R_{ij}^k, a Boolean variable that indicates whether the peer i requests the chunk j from the neighbor k

$$R_{ij}^k = \begin{cases} 1 & \text{if peer } i \text{ requests chunk } j \text{ from neighboor } k, \\ 0 & \text{otherwise.} \end{cases}$$

We now present the core of our chunk scheduling heuristic. Using P_{ij} as defined in (13), we propose the *aggregate priority* Π_i of peer i as a figure of merit for our scheduling algorithm:

$$\Pi_i = \sum_{\substack{j \in M_i \\ k \in N_i}} P_{ij} R_{ij}^k \tag{14}$$

where M_i denotes the set of chunks that peer i requires from the overlay, and N_i denotes the overlay neighbors of peer i. Using this figure of merit, our scheduling problem can be formulated for each peer i as

Maximize: Π_i

Subject to

$$\sum_{j \in M_i} R_{ij}^k \le C_i^k \tag{15}$$

$$\sum_{k \in N_i} R_{ij}^k \le 1 \tag{16}$$

where C_i^k denotes the download capacity of the link between the receiving peer i and its neighbor k.

We proposed a solution of this optimization problem in Bradai and Ahmed (2012a) based on the Hungarian algorithm (Kuhn, 1955) to get the optimal solution.

6. Performance evaluation

After presenting the different component of the video streaming system that we propose, namely the bandwidth allocation and the chunks scheduling mechanism. We present in this section the performance evaluation of our system through simulation using a real P2P streaming systems traces and real video. The simulations were performed using the Matlab-Simulink.

6.1. Simulation setup

In this section we present the network topology, the overlay construction mechanism and the video parameters used in our simulations.

6.1.1. Network topology

We used the BRITE universal topology generator (Medina et al., 2001) to provide a model for the physical network topology. All autonomous systems are assumed to be in the Transit-Stub manner. Each topology consists of 8 autonomous systems each of which has 625 routers. This gives us about 20,000 links in the topology. The download bandwidth of peers varies from 256 kbps to 2 Mbps and is uniformly distributed throughout the network. We introduced sudden bandwidth changes in the network by varying the capacity of peers; this allowed us to observe the effectiveness of our proposed streaming system.

6.1.2. Overlay network construction

In order to construct a randomly connected mesh-based overlay (Magharei and Rejaie, 2006), each peer contacts a centralized directory server (Tracker) for a list of the supplying peers, and randomly selects some of them to establish the neighbor relationship and then maintain a neighbor list. When a peer intends to leave, it sends a message to all its neighbors, notifying the cancelation of it from their neighbor lists.

We emulate peer dynamics based on traces collected from a real-world P2P live streaming system (Liu et al., 2008). When the number of peers rises between two consecutive time intervals, we schedule a corresponding number of peer join events during the interval; when the number of peers decreases, peer departure events are scheduled for a corresponding number of randomly selected peers. Upon arrival, each peer acquires a set of initial upstream peers. The node upload capacities are emulated using values from the traces. The distribution of the peers upload capacity and the portion of their contribution is provided by Liu et al. (2008).

6.1.3. Video parameters

We used in our simulation the standard test video sequence "Bus" at CIF resolution (352 × 288), 30 frames per second, at 500 kbps by using the SVC reference software (JSVM 8.8). We fixed a GOP structure of 32 frames. If not specified otherwise, the stream is composed of 6 layers. A base layer at 200 kbps and each enhancement layer is encoded at 100 kbps. The stream is subdivided into chunks of 1 s.

6.2. Evaluation metrics

We consider the following metrics for the performance evaluation of our proposed streaming system:

- Average Peak Signal to Noise Ratio (PSNR), measured as follows: For each played video we measure the peer PSNR as the average of all layers PSNR, and then we compute the average peer PSNR.
- Upload bandwidth utilization: defined as the ratio of the requested bandwidth over the overall allocated upload bandwidth.
- Useless chunks ratio, defined as the ratio of the decoded chunks to the overall received chunks.
- Number of Rounds to reach the equilibrium.

The chosen metrics reflect the key features of perceived video quality and the effective bandwidth allocation and utilization of the available network capacity.

We generate various network topologies of different sizes, different upstream peers' connectivity degrees (defined as the number of downstream peers connected to the same upstream peer), and different streaming rates.

We use the following function to measure the streaming cost:

$$E_{i,j}(b_{i,j}^k) = \frac{b_{i,j}^k}{x_{i,j} - b_{i,j}^k}$$

where $x_{i,j}$ is the available bandwidth between the downstream peer i and its upstream peer j. This function represents the ratio of the peer i requested bandwidth from peer j, to the residual free bandwidth of link $m_{i,j}$. It expresses the utilization ratio of the link. Indeed, a link with low utilization ratio is considered as a good link since it presents low delay and low bit error rate, because of the low occupation of the intermediate routers' queues. It is worth noting that we choose this streaming cost function as an example to perform our simulation. Any other function to evaluate the streaming cost of a link, satisfying the conditions of convexity and derivability mentioned in Section 4.2.2, can be used.

6.3. Protocols for comparison

First, we compare the performance of our proposed mechanism with two pioneering streaming protocols, namely WU approach

(Wu et al., 2008) and layerP2P streaming protocol (Fig. 10) (Liu et al., 2009). Then, we perform an in-depth evaluation of the performance of our mechanism in terms of the number of rounds to reach the equilibrium and the appropriate network configuration.

We note that LayerP2P adopt the downstream fair bandwidth allocation strategy, where the upload bandwidth is allocated fairly between the downstream peers. That means each upstream peer i splits its bandwidth fairly between all the downstream peers connected to it, i.e.

$$d_j = \frac{U_i}{|L_i|}$$

where d_j denotes the bandwidth allocated to the downstream peer j and L_i is the set of downstream peers of peer i. Furthermore, LayerP2P scheduling mechanism gives more importance to the base layer comparing to the enhancement layers.

6.4. Results and discussion

In order to objectively interpret the video quality results issued from our proposed system and the two others systems, we measure in addition to the videos PSNR, the corresponding utilized bandwidth as well as the ratio of useless chunks. Indeed, the bandwidth utilization metric shows the effectiveness of the bandwidth allocation mechanism in terms of the amount of bandwidth allocated to each peer, while the useless chunks ratio measure the utility of the allocated bandwidth.

6.4.1. Impact of neighbors density

Firstly, we study the impact of neighbor density on the received video quality. We fixed the streaming rate to 1.2 Mbps, and the overlay size to 1000 peers.

Figure 11 represents the variation of the PSNR while increasing the neighbors' density defined as the average number of peer neighbors.

The general trend for the three approaches is the increase of video PSNR with the increase of the neighbors' density. This is because low neighbor density leads to low probability to get the appropriate chunks, mostly for the upper layers that are infrequent in the network due to the layered stream nature. Conversely, increasing the neighbor density leads to increased chance to get the appropriate chunks, and consequently increased the video PSNR.

We note that WU approach performs the worst PSNR among the three approaches. This is due to the fact it does not take into consideration the layers importance and dependency of layers in the layered stream. Indeed, in WU approach an overlay is constructed for each layer. Each layer is requested from an overlay and no coordination is performed between the different overlays. Consequently the amount of decodable received chunks is more and more small.

LayerP2P performs higher PSNR degree since more importance is given to lowers layers in the scheduling process. But, the static downstream peers' fair approach adopted by LayerP2P may

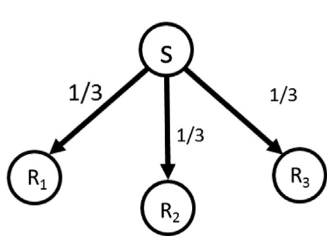

Fig. 10. Bandwidth allocation strategy in LayerP2P.

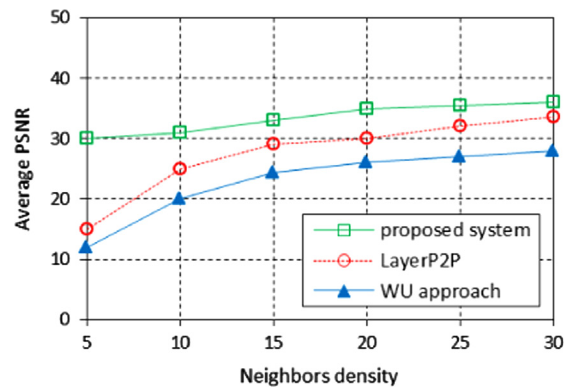

Fig. 11. Average PSNR vs. neighbors' density.

allocate bandwidth to peers that do not need it (peers with low capacities). Consequently, a part of the bandwidth is wasted. In our system the bandwidth allocation is performed taking into consideration the downstream peers' needs in terms of layers. In addition, the upload bandwidth is dynamically allocated, starting with the most important layers (lower layers), then the less important ones (higher layers). Furthermore, our scheduling mechanism requests the lower layers chunks first, and more importance is given to urgent chunks among them. All these justify the higher PSNR values of our system comparing with the two other approaches

In order to investigate in depth the obtained PSNR performance of the three approaches, we measured for the same scenarios, the bandwidth utilization, that means the ratio of the requested bandwidth over the allocated bandwidth, and the useless chunks ratio among the requested bandwidth. The results are represented in Figs. 12 and 13 respectively.

Fig. 12 shows that the WU approach outperforms LayerP2P in terms of bandwidth utilization. We explain this by the fact that parallel auction games are set-up in WU approach to acquire the bandwidth for the different layers, while is LayerP2P, the bandwidth is fairly allocated to peers which may not necessary use it. This is why LayerP2P has the worst bandwidth utilization ratio.

However, when investigating the useless chunks ratio (see Fig. 13), we can see clearly that the useless chunks portion is more important in WU approach since no coordination is performed when requesting bandwidth for different layers. Consequently, even the WU make use of more allocated bandwidth, but the useless requests are higher comparing to P2Player. Thus the video quality in P2Player is better than WU, which is confirmed in Fig. 11. Still our proposed system performs the best bandwidth utilization, thanks to upstream strategy to allocate the bandwidth based on the downstream request. In addition, the useless chunks ratio is the lowest among the three approaches, and this is due to layers incremental bandwidth allocation and the performance of the scheduling mechanism.

Finally, we note that our system provide acceptable video quality (PSNR > 30 dB) even in low neighbors densities (5 neighbors). We remind that a PSNR between 30 and 40 shows that the video quality is of a good enough.

In LayerP2P the quality is acceptable only when the neighbors' density is greater or equal to 20 neighbors. However, the video quality is always less than the threshold 30 dB in WU approach.

6.4.2. Impact of streaming rate

We performed another set of simulation to evaluate the video quality under different streaming rates, varying from 300 kbps to 1800 kbps. We note that the video stream is always encoded in

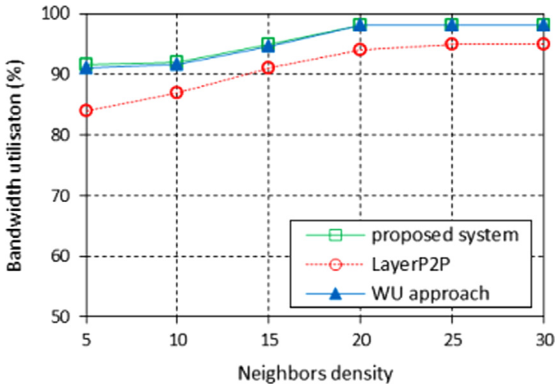

Fig. 12. Bandwidth utilization vs. neighbors' density.

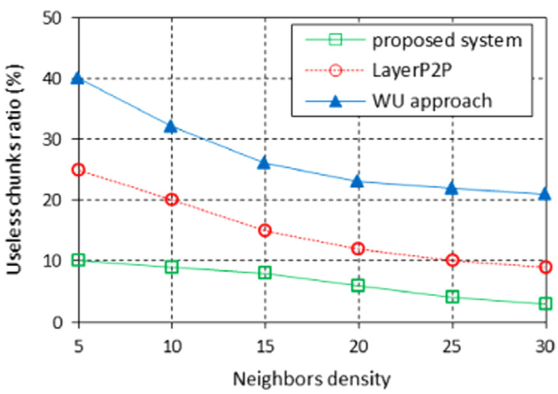

Fig. 13. Useless chunks ratio vs. neighbors' density.

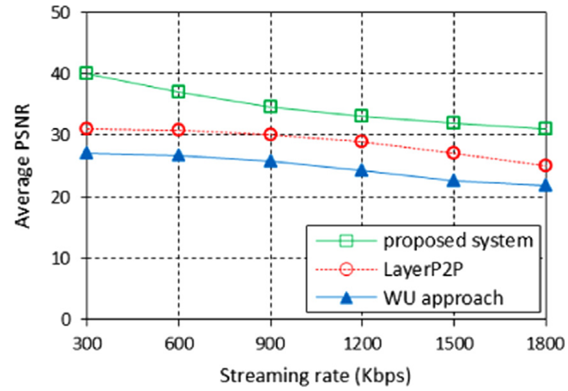

Fig. 14. Average PSNR vs. streaming rate.

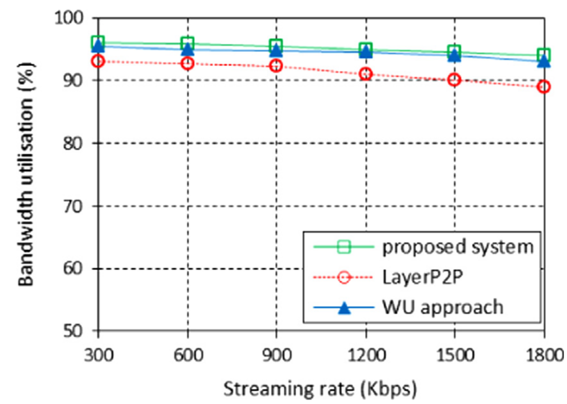

Fig. 15. Bandwidth utilization vs. streaming rate.

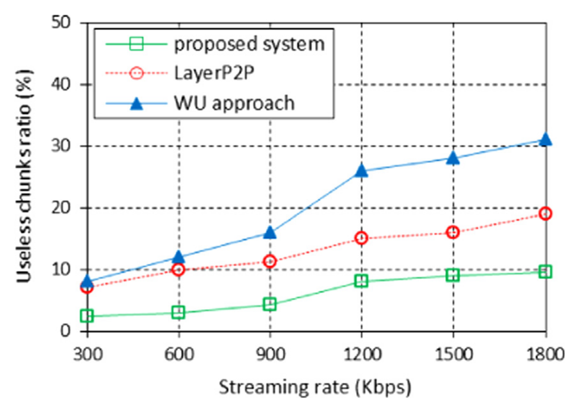

Fig. 16. Useless chunks ratio vs. streaming rate.

6 layers. The base layer rate is encoded at 200 kbps, and all the enhancement layers have the same rate 100 kbps.

For different values of the streaming rate, we measure the average PSNR, the bandwidth utilization and the useless chunks ratio. The obtained results are shown in Figs. 14, 15 and 16 respectively.

In Fig. 14 we depict the average PSNR while varying the streaming rate. As a first observation, we can see that in low streaming rate the videos PSNR is rather of high values comparing with the high streaming rates situation.

Indeed, according to the peers capacity distribution presented in Table 2, for streaming rate of 300 kbps, 90% of the peers in the system (download capacity $\geq$ 300 kbps) can have all the 6 stream layers. This ensures a good proliferation of the layers in the network, and consequently increases the probability to get the requested layers. However, when the streaming rate increases, we observe a drop in video PSNR. This is valid for all the studied strategies. Increasing the streaming rate leads to a reduction in the proportion of high quality level peers in the network. For instance for streaming rate of 1800 kbps, only 6% of peers ($>$ 1500 kbps) can get all the layers.

Consequently the higher layers become increasingly scarce. This means that the probability to get these layers decreases, in turn reducing the video quality level for higher quality level peers.

As a second observation, we can see that the proposed system outperforms the other systems and for all streaming rates situations. Moreover our system delivers an acceptable quality video even in extreme streaming conditions (1800 kbps). We explain this by the fact that it allocates the bandwidth not only following the peers' needs but also respecting the layers dependency, i.e., the most important layers first then the less important ones, which results in high bandwidth utilization (Fig. 15). In addition it allocates the best links to the most important layers. In this way the number of useless chunks is reduced (Fig. 16) and thereby the video PSNR is increased.

6.4.3. *In depth study of the proposed system*

In this section we study in depth the different aspects of our proposed system. First, we study the impact of incentives on the received quality by a peer. Then, we study the impact of peers' capacity on the overall received video quality. Finally, we study the performance of our system in terms of number of rounds to reach the equilibrium in different network configurations.

In the following, for the sake of simplicity, the video is encoded into 3 layers: a base layer l_0 and two enhancement layers l_1 and l_2;

Table 2
Peers' upload bandwidth distribution.

Total upload capacity (kbps)	256	320	384	448	512	640	768	1024	> 1500
Distribution (%)	10	14	9	12	3	2	6	38	6
Contributed upload bandwidth (kbps)	150	250	300	350	400	500	600	800	1000

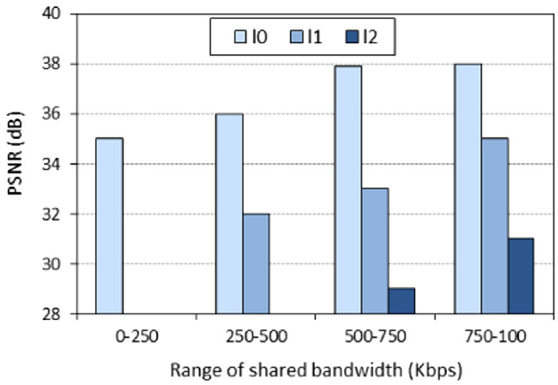

Fig. 17. Average received quality of peers with different shared bandwidth.

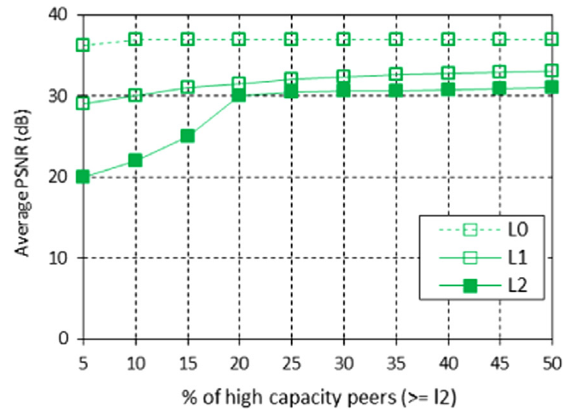

Fig. 18. Average PSNR vs. network configuration.

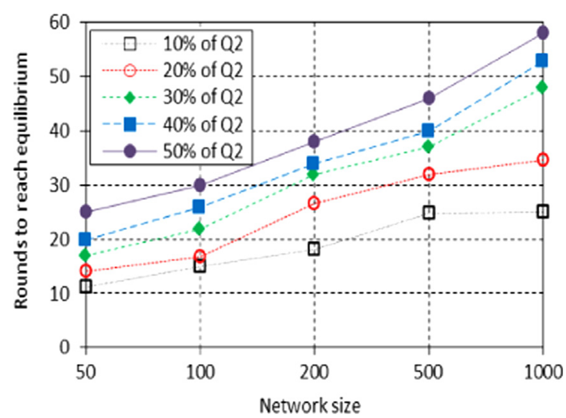

Fig. 19. Number of rounds to reach equilibrium vs. network size.

each of them streamed at 250 kbps (global streaming rate is 750 kbps). The network is composed of 1000 peers; each of them connected to 10 neighbors.

6.4.3.1. Incentives and the video quality. Figure 17 shows the impact of the shared bandwidth by a peer on the video quality that he received, measured by PSNR for each layer. The peers are divided into four groups in accordance with the range of the bandwidth that they shared. It is observed that the received video quality at the peers who shared more bandwidth is obviously better than at peers who shared less bandwidth and this for each layer. The reason is that the more bandwidth the peer shares, the more credit he gets, the best links he gets the bandwidth from. Consequently higher resulting video PSNR. We remember that getting the stream from the best link, means less packet loss ratio, lower delay, etc. which leads to better PSNR value.

We note also that the PSNR of the lower layers is always better than those of the higher layers. This is naturally explained by the incremental bandwidth allocation approach that we adopt. The downstream peer request first the lower layers (from the best links), than the higher layers (from less good links).

6.4.3.2. PSNR vs. peers capacity distribution. We now study the impact of the network configuration on the average PSNR of the received video. We vary the proportion of high capacity peers having the capability to get layer l_2 from 5% to 50%. The goal of this experiment is to investigate the impact of high capacity users in the network on the received quality in our system. The goal is also to identify the best network configuration in terms of peers' capacity in order to ensure a good quality of service for the end users.

In Fig. 18, we can see that the average PSNR of base layer (l_0) and the first enhancement layer (l_1) is always of high values (≥ 29) regardless of the network configuration. This is due to the fact that our proposed system allocates first the lower layers from the best links. In addition, all peers of the network use the lower layers to decode the stream. Consequently the lower layers are more prevalent in the network. However the PSNR of the second enhancement layer l_2 is correlated with the proportion of high capacity peers in the network. At 5% of high capacity peers in the network, the PSNR of l_2 is 20 dBs. This PSNR is acceptable (≥ 30) only when the proportion of these peers in the network is up or equal to 20%. This can be naturally explained by the presence of the higher layers in the neighborhood. A low proportion of high capacity peers results in a scarce presence of the upper layers in the neighborhood, and consequently limited propagation of these layers to the high capacity peers.

6.4.3.3. Number of rounds to reach the equilibrium. We continue investigating network configuration for our streaming system, and we study the number of rounds to reach the equilibrium in varying network size and for different network configuration in terms high capacity peers (noted Q2) ratio in the network.

We observe that the number of rounds to reach the equilibrium increases with the increase in network size, which can be explained by the increase of competition for bandwidth. Moreover, when increasing the number of downstream peers of Q_2 from 20% to 50%, the number of rounds necessary to allocate the bandwidth increases by 40%, while the gain in PSNR of the received video is all most null as shown in Fig. 19.

From Fig. 19, it is evident that increasing the number of high capacity peers in the network leads to significant increase in necessary rounds to reach the system's equilibrium, without real enhancement in the resulting video quality. We conclude also that 10–20% of high capacity peers in the network ensures a good quality level as well as less rounds to reach the equilibrium.

7. Conclusion

This paper introduced a novel bandwidth allocation mechanism for layered video streaming in P2P networks. The proposed mechanism allocates appropriate bandwidth to the appropriate peers while ensuring a minimum quality level to all peers. Each upstream peer organizes a set of auctions to sell its bandwidth, an auction for each layer, starting with the lower layers. In this manner the lower layers are transmitted through the best links, consequently increasing the system throughput.

To study the effectiveness of the proposed mechanism, we performed extensive simulations and we compared the performance of our mechanism with two other approaches. Specifically, we evaluated several essential metrics. First, we evaluated the PSNR of the resulting video. The results show that our proposed mechanism outperforms the two other approaches and ensures always a good video quality (PSNR > 30 dBs) even in low network densities and in extreme streaming rate conditions. Second, we the effectiveness of the proposed incentive mechanism and we study in depth the performance of our proposed system to depict the optimal network configuration for better outcomes.

Based on these findings, we can conclude that bandwidth allocation in P2P streaming systems should be dynamic and should be aware of the needs of neighboring peers, not only in terms of download bandwidth but also in terms of video layers. They should also be aware of the layers' importance, making sure that the most important layers are conveyed over the most reliable links.

As future work, we plan to study techniques to predict the peers' requirements in terms of video quality based on their history, their social relationships, etc. in order to reduce the overhead due to the auction mechanisms.

Appendix. Theorem proof

The basic idea behind the proof is to demonstrate the Nash equilibrium (Chen and Deng, 2006) of each auction A_i^k, and then derive the Nash equilibrium of the whole system. The proof of the Nash equilibrium in A_i^k can be reduced to the proof of the existence of a fixed point for the transfer function describing the evolution of the auction states from an auction round to another. After that we can conclude that our system presents a Nash equilibrium point, which is the set of Nash equilibrium points of all the auctions A_i^k.

We start first by proving the existence of a Nash equilibrium in each auction game A_i^k organized to allocate bandwidth for layer k. Then we derive the Nash equilibrium of the whole system.

The proof of existence of a Nash equilibrium in A_i^k can be reduced to the proof of the existence of a fixed point (Agarwal et al., 2001) for a certain function $\varphi : B \rightarrow B$ where B defines the bidding configuration of our game in a certain round of the auction.

According to *Brouewer* fixed point theorem (Agarwal et al., 2001), a fixed point exists for a continuous function φ of a convex compact set to itself.

(a) From the definition of the upstream peer's bidding strategy (cf. Section 4.2.2), we derive that the set of possible strategies B_i of a player i is a finite set since the price is bounded by $\tilde{p}_i^k$ and the requested bandwidth is bounded by $b_{i,j}^k$. From this we derive that the set of strategies B is compact.

(b) Each biding strategy $B_{i,j}^k \in B$ is defined in $[0, B_i^k] \times [0, \tilde{p}_i^k]$, so the set of strategies B is convex.

From (a) and (b) we conclude that the set of strategies B is a non-empty convex and compact set. Let us now define the function φ and prove its continuity.

The price adjustment algorithm described in Section 4.2.2.2 defines a mapping function $f : P^k \rightarrow P_i^k$ where P^k denotes the matrix of biding prices sent by downstream peers to their upstream peers and P_i^k denotes the vector of price bids of each downstream peer i to its upstream peers: $P_i^k = \{P_{i,j}, \ \forall j \in S_i\}$.

The water-filling algorithm described earlier defines a mapping function $g : P_i^k \rightarrow \beta_i^k$ where β_i^k denotes the set of downstream peer i bandwidth request to its upstream peers: $\beta_i^k = \{b_{i,j}^k, \ \forall j \in S_i\}$

We define the mapping function: $\varphi : (\beta, P) \rightarrow (\beta, P)$ where β denotes the matrix of downstream peer's bandwidth request to its upstream peers. This function maps the bidding configuration of the auction from one round to the following round.

We start by showing the continuity of the function g. Based on the water-filling algorithm, the optimal marginal cost is continuous on P, i.e., for each $P_i^k \in P$ the water filling algorithm associates an optimal marginal cost. In addition the inverse streaming cost function $E_{i,j}^{-1}$ is continuous on P since E is convex and twice derivable. We conclude that the function g is continuous.

Let us now show the continuity of the function f. We can see clearly in the price adjustment strategy of the downstream peer described in Algorithm 2 that the two possible actions for the downstream peer is that for each system bidding configuration either it increases the price $P_{i,j}^k$ or freezes it. From that we derive the continuity of function f. So we show the continuity of f and g, consequently the continuity of φ.

φ is a continuous function from a convex compact set B to itself, so it has a fixed point as given by the *Brouewer* fixed point theorem. Thus, we show the existence of the Nash equilibrium for our auction mechanism.

Since our system is composed of a set of auctions A_i^k to allocate bandwidth for a layer k. And we proved the Nash equilibrium of any of the auctions. We can conclude that our system presents a Nash equilibrium point, which is the set of Nash equilibrium points of all the auctions.

References

Agarwal R, Meehan M, O'Regan D. Fixed point theory and applications. Cambridge, England: Cambridge University Press; 2001.

Boyd S. Convex optimization. New York, USA: Cambridge University Press; 2004.

Bradai A, Ahmed T. On the optimal scheduling in pull-based real-time P2P streaming systems: layered and non-layered streaming. In: Proceedings of the ICC'12. Ottawa; 2012a.

Bradai A, Ahmed T. Differentiated bandwidth allocation in P2P layered streaming. In: Proceedings of the IEEE CAMAD'12. Barcelone; 2012b. p. 110–4.

Bradai Abbas, Abbasi Ubaid, Landa Raul, Ahmed Toufik. An efficient playout smoothing mechanism for layered streaming in P2P networks. Peer-to-Peer Netw Appl 2014;7(2):101–17.

Chen X, Deng X. Settling the complexity of two-player Nash equilibrium. In: Proceedings of the 47th annual IEEE symposium on Foundations of Computer Science, FOCS'06; October 2006. p. 261, 272.

Cheng-Hsin Hsu, Hefeeda M. Optimal coding of multilayer and multiversion video streams. IEEE Trans Multimed 2008;10(1):121–31.

Chun BN, Buonadonna P, AuYoung A, Ng C, Parkes DC, Shneidman J, et al. Mirage: a microeconomic resource allocation system for sensornet testbeds. In: Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors (EmNetsII); 2005.

Coppinger Vicki M, Smith Vernon L, Titus Jon A. Incentives and behavior in English, Dutch and sealed-bid auctions. Econ Inq 1980;18(1):1–22.

Cui Y, Xue Y, Nahrstedt K. Optimal resource allocation in overlay multicast. IEEE Trans Parallel Distrib Syst 2006;17(8):808–23.

Guo H, Lo K, Qian Y, Li J. Peer-to-peer live video distribution under heterogeneous bandwidth constraints. IEEE Trans Parallel Distrib Syst 2009;20(2):233–44.

Habib A, Chuang J. Incentive mechanism for peer-to-peer media streaming. In: Proceedings of the International Workshop on Quality of Service (IWQoS'04); 2004.

Hei X, Liang C, Liang J, Liu Y, Ross KW. A measurement study of a large-scale P2P IPTV system. IEEE Trans Multimed 2007;9(8):1672–87.

Iosifidis G, Koutsopoulos I. Reputation-assisted utility maximization algorithms for peer-to-peer networks. In: Proceedings of the IEEE IWQoS; 2008.

Karunakar AK, Qio Y, Lee B. Scalable video delivery over peer-to-peer network. In: Advanced computing, networking and security. Berlin Heidelberg: Springer; 2012. p. 289.

Kuhn Harold W. The Hungarian method for the assignment problem. Nav Res Logist Q 1955;2:83–97.
Lee I. Reliability analysis of a multiview multi-description video streaming system. In: Proceedings of the 7th international conference on Autonomic & Trusted Computing (UIC/ATC); 2010.
Li B, Qu Y, Keung Y, Xie S, Lin C, Liu J, et al. Inside the new Coolstreaming: principles, measurements and performance implications. In: Proceedings of IEEE INFOCOM. Phoenix, AZ; April 2008.
Liang C, Fu Z, Liu Y, Wu C. Incentivized peer-assisted streaming for on-demand services. IEEE Trans Parallel Distrib Syst 2010;21(9):1354–67.
C. Liang, M. Zhao, and Y. Liu, "Optimal Resource Allocation in Multi-Source Multi-Swarm P2P Video Conferencing Swarms," accepted for publication in IEEE/ACM Trans. on Networking, 2011.
Liu Z, Shen Y, Ross K, Panwar J, Wang Y. Substream trading: towards an open P2P live streaming system. In: Proceedings of the IEEE Conference on Network Protocols (ICNP'08); 2008. p. 94C103.
Liu Z, Shen Y, Ross KW, Panwar SS, Wang Y. LayerP2P: using layered video chunks in P2P live streaming. IEEE Trans Multimed 2009;11(7):1340–52.
Ma R, Lee S, Lui J, Yau D. A game theoretic approach to provide incentive and service differentiation in P2P networks. In: Proceedings of the ACM SIGMETRICS/Performance'04; June 2004.
Ma RTB, Lee SCM, Lui JCS, Yau DKY. Incentive and service differentiation in P2P networks: a game theoretic approach. IEEE/ACM Trans Netw 2006;14(5):978–91.
Magharei N, Rejaie R. Understanding mesh-based peer-to-peer streaming. In: Proceedings of the 16th NOSSDAV'06; 2006.
Manzillo MP, Ciminiera L, Marchetto G, Risso F. CLOSER: a Collaborative Locality-Aware Overlay SERvice. IEEE Trans Parallel Distrib Syst 2012;23(6):1030–7.
Medina A, Lakhina A, Matta I, Byers J. BRITE: universal topology generation from a user perspective. BUCS-TR-2001-003; April 05, 2001.
Park J, Schaar M. Pricing and incentives in peer-to-peer networks. In: Proceedings of the IEEE INFOCOM; 2010. p. 1–9.
Park J, van der Schaar M. A game theoretic analysis of incentives in content production and sharing over peer-to-peer networks. IEEE J Sel Top Signal Process 2010;4(4):704–17.
Qiu D, Srikant R. Modeling and performance analysis of bitTorrent-like peer-to-peer networks. Distrib Inf Syst 2004;34(4):367–78.
Ramzan N, Quacchio E, Zgaljic T, Asioli S, Celetto L, Izquierdo E, et al. Peer-to-peer streaming of scalable video in future Internet applications. Commun Mag 2011;49(3):128–35.
Rückert J, Abboud O, Zinner T, Steinmetz R, Hausheer D. Quality adaptation in P2P video streaming based on objective QoE metrics. Networking. Berlin Heidelberg: Springer; 1–14Networking. Berlin Heidelberg: Springer; 1–14.
SIMULINK. ⟨http://www.mathworks.com/products/simulink/⟩.
Szkaliczki T, Eberhard M, Hellwagner H, Szobonya L. Piece selection algorithm for layered video streaming in P2P networks. Electron Notes Discrete Math 2010;36:1265–72.
Takahashi E, Tanaka Y. Bandwidth allocation by using generalized vickrey auction. In: Proceedings of the Asia pacific symposium on information and telecommunication technologies; 2001.
Wang W, Li B. Market-based self-optimization for autonomic service overlay networks. IEEE J Sel Areas Commun 2005;23(12):2320–32.
White LR. Effective marginal costing: know your resource needs. Resour Consum Account Inst 2010, ⟨http://www.rcainstitute.org/rcadocuments/Present.AGA.Marginal%20Costing.3.2010.ppt⟩.
Wu C, Li B, Li Z. Dynamic bandwidth auctions in multi-overlay P2P streaming with network coding. IEEE Trans Parallel Distrib Syst 2008;19(6):806–20.
Zhang M, Xiong Y, Zhang Q, Yang S. Optimizing the throughput of data-driven peer-to-peer streaming. IEEE Trans Parallel Distrib Syst 2009;20(1):97–110.
Zhang X, Liu J, Li B, Yum TP. CoolStreaming/DONet: a data-driven overlay network for P2P live media streaming. In: Proceedings of the IEEE INFOCOM'05. Miami, USA; 2005.
Zhang Y, Zhou X, Song J, Yue Y, Liao D. Time-stamped equal size segmentation and chunk scheduling algorithms for SVC based P2P streaming systems. In: Proceedings of the IEEE 18th International Conference on Parallel and Distributed Systems (ICPADS); 2012. p. 706–7.
Zhao B, Liu J, Chiu DM. A mathematical framework for analyzing adaptive incentive protocols in P2P networks. IEEE/ACM Trans Netw 2012;20(2):367–80.
Zhou L, Wang X, Li Y, Zheng B, Geller B. Optimal scheduling for multiple description video streams in wireless multihop networks. IEEE Commun Lett 2009;13(7):534–6.