



HAL
open science

Avoiding Quality Bottlenecks in P2P Adaptive Streaming

Samir Medjiah, Toufik Ahmed, Raouf Boutaba

► **To cite this version:**

Samir Medjiah, Toufik Ahmed, Raouf Boutaba. Avoiding Quality Bottlenecks in P2P Adaptive Streaming. IEEE Journal on Selected Areas in Communications, 2014, 32 (2), pp.734 - 745. 10.1109/JSAC.2014.140406 . hal-00965122

HAL Id: hal-00965122

<https://hal.science/hal-00965122>

Submitted on 25 Mar 2014

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

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

Avoiding Quality Bottlenecks in P2P Adaptive Streaming

Samir Medjah, *Member, IEEE*, Toufik Ahmed, *Member, IEEE*, and Raouf Boutaba, *Fellow, IEEE*

Abstract—This paper addresses the problem of quality bottleneck in adaptive SVC streaming. Quality bottleneck occurs in adaptive streaming systems when the desired video quality cannot be obtained even if the network capabilities are sufficient. In the context of SVC layered video streaming, we have observed that enhancement layers remain around the video source and fail to reach all the participating peers. To overcome this problem, we propose an adaptive SVC streaming solution that cooperatively integrates strategies of overlay formation, data scheduling and content adaptation. Performance evaluation using simulations shows that the proposed streaming solution reduces the quality bottleneck, increases churn-tolerance and optimizes bandwidth utilization.

Index Terms—adaptive streaming, layered video, SVC, overlay networks, P2P, quality bottleneck, data scheduling, quality smoothing

I. INTRODUCTION

THE LAST few years witnessed tremendous deployments of audio and video streaming systems. These deployments are based on a variety of architectures, including: Peer-to-peer (P2P) overlays such as Spotify [1], PPLive [2], and Zattoo [3]; cloud platforms such as CloudStream [4]; Content Distribution Networks (CDN) such as Netflix [5] and YouTube [6]; and other proprietary solutions. To a large extent these deployments realize what traditional streaming and multicasting architectures have struggled to achieve large-scale distribution of video-on-demand and live video services to a massive number of end-systems distributed over the Internet.

A. Motivations

CDN and Cloud architectures are extensively used nowadays for delivering video services at large scale. However, they achieve so at the cost of overprovisioned network infrastructures and at growing concern of network operators. P2P streaming solutions constitute a viable alternative as they offer a better tradeoff between the utilization of the underlying network infrastructure and the delivery of overlay services. However P2P networks present a number of challenges yet to be resolved [7]. Most notably, the ad hoc nature of P2P networks makes them vulnerable to: (1) individual peer behavior such as random departure and

arrival (a.k.a. peer churn), asymmetric contribution in terms of network and storage resources (a.k.a. free riders), etc.; and (2) collective behavior of peers such as massive arrival or departure (a.k.a. flash crowds). These behaviors result in highly dynamic overlay topologies and load unbalances, and consequently in degraded services, especially for real-time applications with stringent delay, packet loss and bandwidth requirements. Peer churn, for instance, forces neighboring peers to find new neighbors and establish new connections with them. This introduces significant delays and a high probability of service discontinuity.

P2P video streaming applications have tackled the unpredictability of peers' behavior using layered video content since its playback is possible with partial data. Examples of layered video formats include MDC (Multiple Description coding) [8] and SVC (Scalable Video Coding) [9], both use layered encoding schemes to facilitate content adaptation, delivery and playback in the events of changing, unstable, or unexpected network conditions. By using layered video content, it has been shown that peer churn can be tolerated to a certain degree [10]. But layered video content has also other advantages, including the ability of heterogeneous peers to consume tailored content that meets their different terminal and network capabilities as well as their preferences (e.g., frame rate, resolution, video quality, etc.) [9].

However, streaming continuity is achieved at the cost of decreased quality of the received content (i.e., playback with partial data). In addition, the decomposition of the video stream into incremental and dependent layers such as in SVC raises the issues of data distribution and scheduling [11] which need to be carefully addressed. In non-layered streaming over P2P networks, peers download any available chunk, whereas in layered streaming, and due to layer dependency, peers are more concerned with the decision of which chunk of which layer to download. In this context, efficient scheduling of the data chunks from neighboring peers will determine the video quality level that every peer will be able to get from the overlay or contribute to it [11][12][13][14][15]. Data chunks scheduling in adaptive streaming over P2P networks is subject to several considerations such as the dependency relationship between multiple representations of the video (e.g., SVC layers), the links capacities between the peers, the urgency of the data chunks in case of live video streaming, or whether the data chunks are pushed by the sender, requested by the receiver or a combination of both.

Data chunks scheduling has also to take into consideration the need for adaptive streaming in order to provide a stable long-term quality. Indeed, even if the video content

Manuscript received April 15, 2013; revised September 22, 2013.

S. Medjah and T. Ahmed are with CNRS-LaBRI/Univ. de Bordeaux, 351, Cours de la Liberation, F-33405, Talence Cedex, France (e-mail: medjah@ieec.org; tad@labri.fr).

R. Boutaba is with School of Computer Science, University of Waterloo, 200 University Ave. W., Waterloo, ON, Canada. (e-mail: rboutaba@uwaterloo.ca).

Digital Object Identifier 10.1109/JSAC.2014.140406.

can be adapted on the fly and switched from one quality representation to another, the user quality of experience can be unstable due to frequent video quality changes and sharp move between qualities. Smooth video adaptation [16][17] aims at providing a stable quality of experience. Smoothing can be obtained by maintaining the same quality level during a predetermined period of time before considering a quality level enhancement or a progressive quality level degradation. Further, it is worth noting that the availability of the desired data chunks in the neighborhood determines how the overlay network is built and maintained. Indeed, in a scenario where a newly joining peer is connected to other peers providing only a low quality video level, this new peer will be constrained to this quality level and shares it back with other peers even if its capabilities permit a higher quality level.

As illustrated in figure 1, inefficient bandwidth utilization occurs when peer (A) with high uplink capacity is attached to peer (B) with high download capacity, but the maximum delivered quality that (A) receives and distributes back is far less than the link capacity. Also, a video quality loss occurs when peer (A) with high uplink capacity is attached to peer (B) with low download capacity since peer (B) will only get and serve back lower quality video streams. To avoid such loss, peer (A) needs to serve other peers to fully utilize its upload capacity.

Since peers contribute to the overlay with the content that they acquire from the network, they need to be carefully connected to other peers in order to avoid what we call the “quality bottleneck” problem. The quality bottleneck problem is more severe in case of high peer churn.

B. Contributions

In this paper, we identify the problem of quality bottleneck in the context of adaptive video streaming in overlay networks, as the “fading and vanishing of the video quality level obtained by some peers in the overlay even if available resources allow a higher quality”. We believe that an efficient solution to the quality bottleneck problem must jointly address the following three aspects:

- a) **Overlay Formation:** Overlay formation concerns the process of connecting newly joining peers to the existing peers. It comprises the mechanisms used to adapt the neighborhood of a peer subject to neighbors’ departures or in order to augment the current neighborhood. This is perhaps the most important task when building a P2P streaming system. Indeed, the neighborhood assigned to a peer will heavily impact the data distribution, which depends on the availability of the data chunks in the neighborhood, and the capacities of the links between a peer and its neighbors.
- b) **Data Distribution:** The data distribution or scheduling involves the mechanisms used to acquire and redistribute the data chunks of the content being shared. Data acquisition can be achieved in pull [14], push [17] or push-pull [18] modes. In pull-mode, the receiver peer is responsible for requesting the desired data from its neighbors. In push-mode, the data transmission is controlled by the sender peer, and the receiver peer

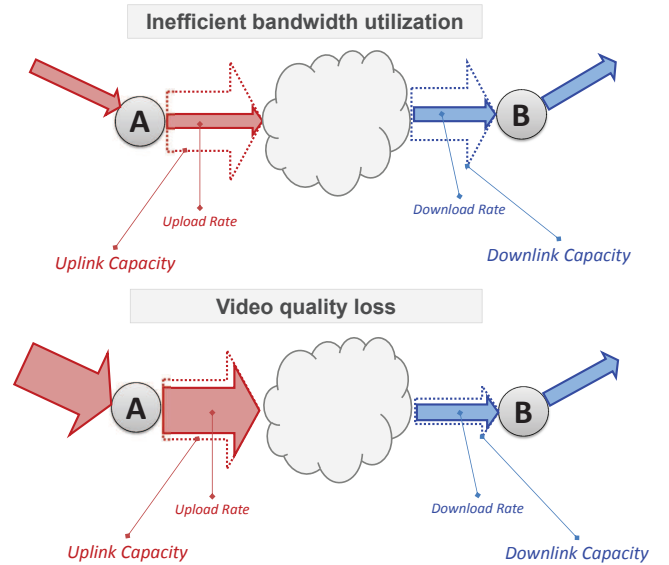


Fig. 1. Connection leading to inefficient bandwidth utilization

will obtain data from the sender peer without prior requests. Finally, in push-pull mode, important data may be pushed without prior requests (e.g. the base layer in the case of SVC video), while other enhancement data can be requested by the receiver peer (e.g., enhancement layers). Regardless of whether the data is distributed in pull or push mode, the data chunks need to be scheduled:

- In time: the data chunks must be acquired before their consumption date or playback deadline.
- Through the peer’s neighbors: the data chunks need to be acquired from (respectively distributed to) relevant peers.

The problem of data scheduling can be tackled at the receiver side [19], at the sender side [20], or at both sides [21]. The data distribution process also needs to address potential transmission errors using mechanisms such as forward error correction [22], or network coding [23][24].

- c) **Content Adaptation:** Content Adaptation is essential in the context of dynamic and adaptive service provisioning. In video streaming systems, it has been shown that the video playback continuity and the absence of frame freezing events are as important as the video quality level. Indeed, quality of experience (QoE) studies [25][26] have shown that the user is more likely to prefer to trade video quality for streaming continuity (absence of stalling events) when facing unstable and varying network conditions. Other studies [27][28] have shown that selecting the highest quality level is not optimal for some video content types. Thus, these subjective studies must be considered for an efficient adaptive streaming system.

These three aspects must be addressed jointly in order to avoid the quality bottleneck problem. Indeed, a good technique for overlay organization may connect peers between them in an optimal way regarding their capabilities, but an inappropriate scheduling technique may fail to achieve efficient data chunks

scheduling. Similarly, a good technique for scheduling that does not take into account how the peers are being connected among them will not perform better. Finally, even if overlay formation and data scheduling are jointly addressed, peers may successfully achieve a better throughput but will fail to achieve a better quality of experience due to the adaptability characteristic of layered video. Typically, a content adaptation technique should drive the data chunks scheduling technique and this technique needs to be aware of the overlay formation used in the system. Towards this goal, we develop in this paper, a comprehensive solution for efficient adaptive streaming over P2P networks. Specifically, we propose an adaptive SVC streaming solution that cooperatively integrates strategies for overlay formation, data scheduling and content adaptation.

C. Paper Organization

The remainder of this paper is organized as follows. Section II surveys some research works found in the literature that address the joint overlay construction and data scheduling. Section III presents background knowledge and definitions. Section IV describes our proposed solution for adaptive streaming, and section V discusses its performance evaluation. Finally, section VI concludes this paper.

II. RELATED WORK

Significant research progress has been made in the field of pull-based peer-to-peer layered streaming [11][14][19]. The main objective of these systems is to provide the peers with the highest video quality. However, even though a lot of techniques have been proposed in the literature for data scheduling in layered video streaming, only a few have focused on joint overlay formation and scheduling for efficient layers delivery. For instance, authors in [29] have defined a classified P2P overlay scheme for SVC streaming (*CLS*) where the overlay topology is composed of different groups. Peers with similar uploading bandwidth are put into the same group and communications between peers are only possible within the same group. Such solution can be inefficient because of dynamic arrival and departure of peers resulting in inefficiently formed groups. The authors of [30] proposed a bandwidth-aware hybrid overlay architecture for push-based P2P streaming of scalable video. The overlay is also organized into hierarchical clusters but with limited capacity in terms of number of peers. Within each cluster, tree overlays are formed for data exchange using a push-based delivery. The distribution trees are constructed while taking into account peers bandwidth in order to place more capable peers near the video source. However, this solution suffers from a high cost relative to tree construction and maintenance and its inability to scale to a large number of users.

Using a hybrid tree-mesh overlay, authors in [18] proposed LayeredCast, which aims at combining the advantages of both overlay organizations (i.e., mesh and tree). The base layer is pushed into the tree overlay while enhancement layers and base layer retransmission are acquired from the mesh overlay using a pull-based delivery. The authors also proposed a new heuristic for receiver side data scheduling. Even though LayeredCast combines data scheduling and random overlay

formation, the unpredictability of the peer organization and the maintenance cost of the base layer tree make the solution not optimal for preventing the occurrence of quality bottlenecks.

III. BACKGROUND

In this section, we describe some background knowledge related to adaptive SVC streaming. First, we present the SVC video format in terms of layers composition and bitstream structure. Second, we discuss the need for metadata structures in order to describe the SVC bitstream for data scheduling. Finally, we introduce the problem of SVC description update and advertisement in the context of live streaming.

A. SVC Video Format

Layered adaptive streaming is supported by the H.264/SVC format [9] which allows a video stream to provide various representations of the same video. Furthermore, it offers what we call the “*encode once, decode many*” feature, i.e., the video is encoded into a single file but can be decoded in different ways in order to have different representations (different frame size / frame rate / picture fidelity). To support such feature, SVC format introduces three types of scalability:

- *Spatial scalability*: different frame resolutions.
- *Temporal scalability*: different frame rates.
- *Quality scalability*: different compression levels.

Any combination of the above three scalabilities determines the quality of the resulting video representation. The combination having the lowest frame resolution, the lowest frame rate, and the lowest compression level is called the Base Layer (BL) while the other combinations are called the Enhancement Layers (ELs). Due to SVC compression algorithm, the higher layers depend on lower layers for decoding. Indeed, the SVC format has also let to new compression techniques to allow efficient storage of the encoded representations [26]. SVC format has the ability to allow the decoder/player to switch from one representation to another by simply adding or dropping enhancement layers during the video playback.

As an extension to AVC, SVC relies on the principle of Network Abstraction Layer for organizing the video bitstream. SVC bitstream is a collection of atomic data pieces named Network Abstraction Layer Units (NALUs or NALs). There is one type of NALs that is very important for informing the application about a complete description of the layers. This unit is called SEI NAL (*Supplemental Enhancement Information*). It contains information such as the number of layers, the dependency relationship between the layers, the number of scalabilities and their description.

Another important NAL unit is called the IDR NAL unit (*Instantaneous Decoding Refresh*). This NAL represents a synchronization point within the overall stream since it allows the decoder to switch to another representation as no backward dependency exists after an IDR NAL.

In Summary, SVC adaptation relies on these IDR NALs to perform on-the-fly layer switching. Figure 2 illustrates an example of 6-Layers SVC bitstream composition. In this figure, we can see a sequence of subsequent NALs composing the SVC bitstream; each NAL unit belongs to a specific layer (0 to 5). IDR NALs are highlighted.

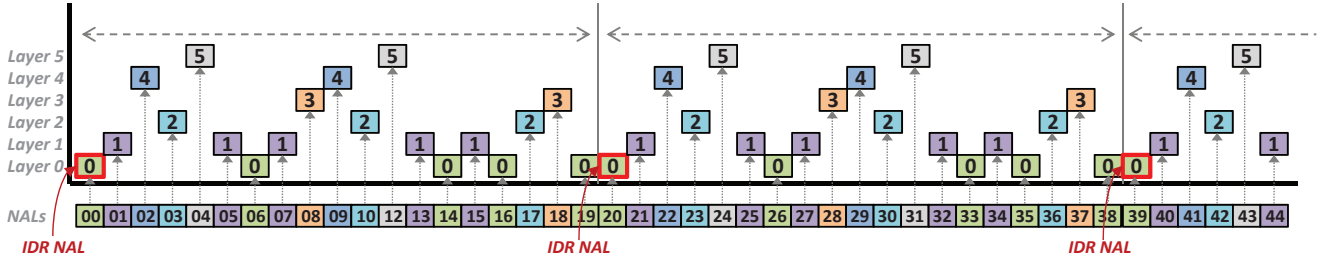


Fig. 2. Example of a 6-Layer SVC Bitstream Composition

B. SVC Video Content Distribution

In order to ease the distribution of SVC video content and to allow efficient adaptive streaming, the SVC stream needs to be described using a high level metadata structure. Such structure may contain general information about the stream such as its title, duration, and spatial resolution.

For the sake of the data chunk scheduling, the metadata provides information about the binary composition of the video stream. This information, referred to as the StreamMap, describes the data chunks to be exchanged.

The StreamMap can have a simple structure. In the case of non-layered video using MPEG2-TS video, the StreamMap includes just two indications: (1) the offset of the MPEG2-TS packets (multiple of 188 bytes) and, (2) the number of the packets to be retrieved. However, the StreamMap is detailed further in the case of layered video format (e.g., SVC).

In this case, it contains information on NALs with their different characteristics (Size, Type, Layer ID, etc.) which are needed in order to achieve an efficient data scheduling and to enable dynamic content adaptation (i.e. SVC layer adding/dropping).

In our solution, we propose an XML-based Resource Description file (rdx file) in order to describe any exchanged SVC video content whether it is for live or on-demand streaming. This description format is designed primarily for H.264/SVC but it is also capable of supporting H.264/AVC streams. Moreover, it can be easily extended to other codecs. Manifest files describing multimedia content for HTTP adaptive streaming have been used in industrial software such as:

- ISM/ISMC manifest file in Microsoft HTTP Smooth Streaming [31],
- F4M manifest in Adobe Dynamic HTTP Streaming [32],
- M3U8 file in Apple HTTP Adaptive Streaming [33],

MPEG-DASH does also introduce a manifest file dubbed as Media Presentation Description (MPD)[34][35]. However, all these technologies are designed mainly for HTTP streaming. Thus, these manifest files lack of a fine grain presentation of the SVC content at the NAL level. Our RDX file embeds both general information about the video stream and the StreamMap. An example of the manifest file structure is shown in figure 3.

Each NAL unit is described with the following information:

- *ID*: The NAL unit identifier which is the index of the NAL within the full SVC stream.

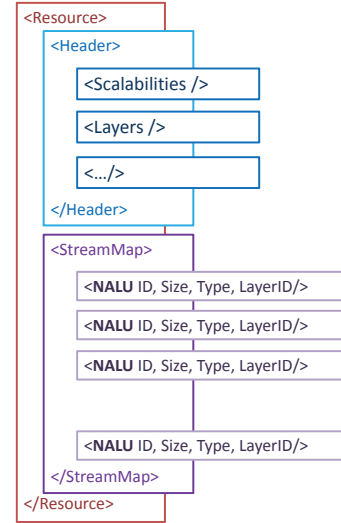


Fig. 3. Example of an RDX File Structure for SVC Stream

- *Type*: The type of the NAL unit as described in the SVC extension to H.264/AVC standard.
- *Layer*: The ID of the layer to which the NAL unit belongs to, if applicable. (The layer ID is irrelevant in the case of non-VCL NAL units)
- *Size*: the total size of the NAL unit in bytes.

Based on this information, SVC adaptive streaming can be achieved by planning the targeted layers set according to the consumer context (network/terminal capabilities, and user preferences). A fine-grained video adaptation can be further performed by means of an efficient data chunks scheduling.

C. SVC Description Advertisement/Exchange

In the case of pull-based streaming, the consumer peer needs to acquire the video stream description before requesting the actual video content. For on-demand streaming, the stream description (rdx file) can be generated offline, once the video stream is available. The manifest file can be transmitted at the beginning of the session and the scheduling of the different data pieces can then be optimally achieved. For live streaming, the manifest file transmitted at the beginning of the session can only contain general static information about the video stream such as title and supported layers. The StreamMap can only be transmitted incrementally and periodically as the video stream is being generated at the source.

streaming avoiding the quality bottleneck problem. Our approach consists in jointly optimizing overlay formation, data distribution and content adaptation. Specifically, we present here our proposed SVC-enabled data scheduling strategy along with our overlay formation mechanism and SVC adaptation technique, and how they are used to maximize the perceived quality of experience while avoiding the quality bottleneck. Our solution relies on the following assumptions:

- The peers are organized in a pull-based mesh topology,
- A source (or a set of sources) provide(s) an SVC video stream with different scalability levels.
- Video is composed of different SVC layers $\{L_0, L_1, L_2 \dots L_K\}$ with different bitrates $\{R_0, R_1, R_2 \dots R_K\}$. For the sake of simplicity, each layer of the video stream depends on all previous layers, i.e., layer L_n depends on layer L_{n-1} . with L_0 being the base layer.
- A peer P_i has the following attributes:
 - U_i : overall uplink capacity,
 - D_i : overall downlink capacity,
 - $V_i(t)$: the viewed layer by the peer at instant t ,
 - $O_i(t)$: the out-degree of the peer i at instant t ,
 - $I_i(t)$: the in-degree of the peer i at instant t ,
- A peer equally distributes its uplink capacity to serve all neighbors that consume its content.
- A peer's session lifetime within the system can be described by statistical distributions (*Exponential*, *Weibull*, *Log-Normal*, etc.) [38][39][40].
- The peer arrival distribution can be described by the mean of statistical distribution (e.g. Exponential).
- There is a central entity (i.e. a Tracker) orchestrating the overlay formation, by providing the list of neighbors to each newly arriving peer.

Considering the above assumptions and system model, our goal is to find an overlay formation scheme along with a scheduling strategy that allows a better delivery of a relatively high and stable quality to each peer during the entire lifetime of its presence in the overlay.

A. Scheduling Strategy

Considering the layers availability in the neighborhood, and the capacity of the links connecting these neighbors, the scheduling strategy aims to obtain the higher quality of the video while considering the eventual departure of each neighbor and its impact on the received quality. The main idea behind our proposed strategy is to dispatch data requests among the peers according to what they can provide and proportionally to their upload capacity.

We consider a neighborhood composed of M Peers: P_1, P_2, \dots, P_M . Each peer P_i is providing U_i of all its available layers $\{L_0, L_1, \dots, L_{K_i}\}$ as follows:

- x_{ij} is the amount of data of layer L_j that will be requested from peer P_i .
- a_{ij} is a boolean variable which indicates if layer L_j is available from peer P_i .
- Regarding the downlink capacity D_P of the local peer P , the higher layer that can be obtained by the peer is L_{MAX} such that: $R_{MAX} \leq D_P$.

- The higher layer that a peer is effectively able to obtain from its neighborhood can be determined by :

$$L^* \leq L_{MAX}$$

$$L^* = \text{argmax}_{a_{ij}} (\sum_{i=1}^{i=M} a_{ij} \cdot x_{ij} \geq R_j; j \in \{0, 1 \dots a\})$$

subject to:

$$\sum_{j=1}^{j=K_i} a_{ij} \cdot x_{ij} \leq U_i; j = 1 \dots M$$

$$\sum a_{ij} \cdot x_{ij} \leq D$$

These constraints can be interpreted as follows:

- The overall requested data from a peer P_i must not exceed its uplink capacity U_i that is providing.
- The overall requested data from all the neighbors must not exceed the downlink capacity (D) of the peer.

In order to solve this optimization problem, we must find the matrix $X = (a_{ij} \cdot x_{ij})$.

Determining the highest quality that can be obtained from a certain neighborhood, under the above constraints is known to be NP-Hard [36]. To solve it, we devise a heuristic solution called the Themis Heuristic. The goal of our heuristic is twofold: on one hand, the heuristic tries to find the higher possible layer, and on the other hand, it tries to spread the requests among the neighbors according to their uplink contribution.

The proposed scheduling strategy depends on the neighborhood proposed to a newly joining peer. This strategy can become less effective if the neighborhood does not have the data that interests the new peer. In the following sub-section we present our strategy for the overlay formation in order to avoid the quality bottleneck problem.

B. Overlay Formation Strategy

1) *Stable neighborhood*: During the construction of the overlay topology, it is important that the neighborhood proposed to the new peer is stable during the peer presence within the system. A stable neighborhood avoids the situations where the peer has to look for new neighbors in case of churn. In order to determine a stable neighborhood to propose to a new peer, we study the lifetime duration of a stable neighborhood.

In this study, we consider an overlay where leaving neighbors are not replaced. In figure 9, we show an example of a newly joining peer that has been connected to 4 neighbors. At the beginning of the session, the local peer receives data up to $U_1 + U_2 + U_3 + U_4$ from its neighborhood. After a certain period, neighbors will leave the overlay respectively at T_1, T_2, T_3 , etc. (peers are numbered according to the order of their departure). Here, as shown in figure 9, \mathcal{R}_i represent the residual lifetime of neighbor peer P_i . Let \mathcal{L} be the session lifetime of the local peer. For memoryless distributions, \mathcal{L} and \mathcal{R}_i follow the same distribution (*the waiting time paradox*).

While connected to such neighbors, the newly joining peer will be able to have a stable quality level from its neighbors, with respect to data availability, as long as T_1 (the departure time of the first neighbor) is greater than the average peer's lifetime $E[\mathcal{L}]$. The probability of such state is $(T \geq E[\mathcal{L}])$, where $T = \min(T_1, T_2, \dots, T_M)$.

Get_Max_Possible_Layer()

```

1:  $L^* = L_{MAX}(D)$  //max layer subject to DL capacity
2:  $k = L^*$  //trying to acquire the higher possible layer
3: while ( $k \neq 0$ )
4:    $U' = U$  //backup the peers uplink vector
5:   if ( $\sum_{i=1}^{i=M} a_{ik} U_i \geq R_k$ ) //enough data to get layer k?
6:      $\forall i : x_{ik} = \frac{a_{ik} U_i}{\sum_{i=1}^{i=M} a_{ik} U_i} R_k$  //req load balancing
7:      $\forall i : U_i = U_i - x_{ik}$  //update peers uplink vector
8:      $k = k - 1$  //proceed with lower layer
9:   else
10:     $\forall j \geq k, \forall i : x_{ik} = 0$  //abandon layer k, and higher
11:     $U = U'$  //restore peers uplink vector
12:     $L^* = L^* - 1$  //target a lower layer
13:     $k = L^*$  //re-start from this new targeted layer
14:   endif
15: endwhile
16: return  $L^*$ 

```

Fig. 8. Scheduling Strategy (Themis Algorithm)

In the case where \mathcal{L} follows an exponential distribution $\mathcal{L} \sim \mathcal{E}(\lambda)$, we can obtain:

$$P(T \geq t) = P(\min(T_1, T_2, \dots, T_M) \geq t)$$

$$P(T \geq t) = P(\bigcap_{i=1}^{i=M} T_i \geq t)$$

$$P(T \geq t) = \prod_{i=1}^{i=M} e^{-\lambda_i t}$$

$$P(T \geq t) = e^{-\Lambda t}; \Lambda = \sum_{i=1}^{i=M} \lambda_i$$

In the case where all \mathcal{L}_i are identically distributed and having the same parameter λ , we have:

$$\Lambda = \sum_{i=1}^{i=M} \lambda_i = M\lambda$$

Then:

$$P(T \geq t) = e^{-M\lambda t}$$

$$P(T \geq E[\mathcal{L}]) = e^{-M\lambda \mathcal{L}}; \text{ knowing that } E[\mathcal{L}] = \frac{1}{\lambda}$$

$$P(T \geq E[\mathcal{L}]) = e^{-M}$$

The analysis of this probability permits us to tune the neighborhood size according to the characteristics of the node lifetime distribution in order to ensure a stable and churn-tolerant neighborhood. Based on the desired probability value, the size of the neighborhood M can be set accordingly. Such analysis can be easily extended to other peer lifetime distributions (Log-Normal, Weibull, etc.).

2) *Preferential attachment*: Once we have determined the size of the stable neighborhood able to provide the desired quality, we propose to organize the new peers into a modified scale-free network [41] using preferential attachment procedure according to the SVC video layer being delivered by the peers. The original preferential attachment as defined in [41] attaches the newly joining peers according to the serving capability of a peer. The serving capability is defined as the ratio of the uplink capacity of a peer to its out-degree. Thus,

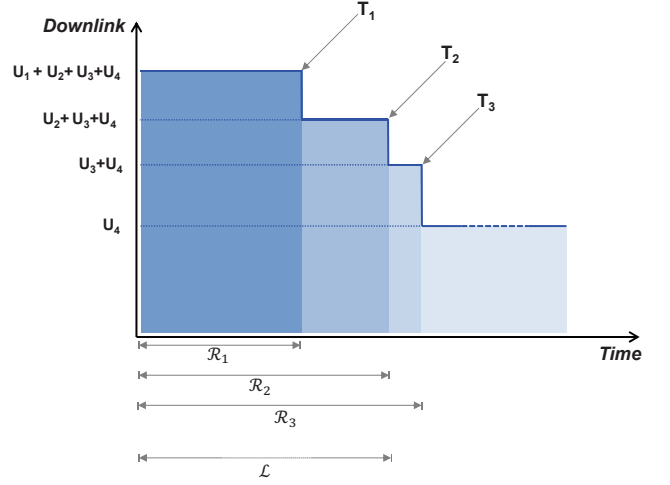


Fig. 9. Stable Neighborhood Lifetime

peers with higher serving capability are attached to more peers. In this paper, we propose to adapt this preferential attachment to include the similarity of the peers in terms of the quality level a sender peer can provide and the quality level a receiver peer can obtain. The preferential attachment π_i to a peer P_i is thus expressed as:

$$\pi_i = P(\text{LinkToPeer}_i) = \left(\frac{\alpha_i \beta_i}{\sum_{j=1}^{j=N} \alpha_j \beta_j} \right)$$

where:

$$\alpha_i = \frac{U_i}{O_i(t)}$$

$$\beta_i = \frac{1}{\sqrt{(D - RV_i(t))^2 + (U - RV_i(t))^2}}$$

The probability π_i of connecting a new joining peer P to a peer P_i , which is already part of the overlay, takes into account the two following aspects:

- Peers are attached according to their contribution to the overlay, in such a way that peers having high serving capacity attract the newly joining peers. Conversely, saturated peers will repel the joining peers. This aspect of the attachment is expressed by the α part in the preferential attachment formula.
- Peers are also attached according to the higher layer that they can serve to other peers. This aspect is represented by the β part in the preferential attachment formula. It aims to connect peers having similar viewing layer and thus similar video quality representation. It can be seen as the minimization of the difference between the currently viewed layer by the peer and the capacity of the joining peer in order to get and to distribute this layer back to other peers. This condition is essential for mitigating the quality bottleneck problem.

The preferential attachment procedure can be essentially described as follows. When a new peer joins the overlay, the probability of its attachment to a random peer of the overlay is proportional to the contribution of this peer in terms of uplink capacity and layer availability. Therefore, we use the following procedures to build such an overlay:

- a) **Peer join:** when a peer P_i joins the overlay network at time t , the central control entity sorts the existing peers according to the preferential attachment probability and connects the new peer to the M peers having the highest probability of attachment. Thus, the new peer starts with a zero out-degree and an in-degree equal to M .
- b) **Peer departure:** when a peer leaves the network, all of its neighbors lose a download link, and their download rates decrease accordingly. In order to compensate for this decrease, a new download link is required for each of these peers. The node replacement process is similar to the Peer join process, except in this case, only one peer (i.e. with the highest probability of attachment) will be chosen.

C. Streaming Adaptation Strategy

Even with an optimal overlay formation, the performance of the data scheduling can still be affected by varying network conditions. For example, bandwidth fluctuations can lead to unstable video quality, and possibly streaming discontinuity (i.e. stalling events). In the case of pull-based streaming, the receiver peer has to decide which quality level to choose. To this end, authors in [16][17] proposed a smoothing function that decides which layer to request in order to guarantee a stable quality for a certain period of time, referred to as the smoothing window. To obtain a stable quality, the proposed smoothing function tries to minimize not only the number of layer changes (i.e. frequency reduction) but also the size of jumps between two consecutive layers (i.e. amplitude reduction). In our solution, we have used the smoothing function in order to steer the data chunks scheduling task so as to overcome bandwidth fluctuations and achieve a significant enhancement of the user's perceived quality of experience. Figure 10 provides an overview of the different system's components and their inter-working. We see that the playout buffer is composed of different regions:

- a) **Region 1 - Decoding buffer:** this region contains all the downloaded data which is ready to be fed to the decoder.
- b) **Region 2 - Request buffer:** in this region, the peer has already acquired the data description (StreamMap & BufferMaps). Thus, the peer can form requests to obtain these chunks.
- c) **Region 3 - Fetching buffer:** in this region, the peer does not know the composition of the stream so it has to request first the stream description corresponding to this region.

After the current playhead position, we find the already acquired data (Region 1) that is ready to be supplied to the decoder. Beyond this region is the part of the stream to be downloaded and for which the bitstream description has already been acquired (Region 2). Finally, the third region (Region 3) describes portion of the stream for which both the bitstream description and the actual data need to be acquired. In figure 10, we also show the SVC smoothing and scheduling functions:

- a. First, the SVC smoothing function takes as inputs the history about viewed layers and the download bandwidth performance and decides which SVC layer to target

for the next stream portion, according to the algorithm described in [16][17].

- b. On the basis of the new stream portion's description from Region2, the target SVC layer decided by the smoothing function, and the neighbors' BufferMaps, the scheduling function (i.e. the Themis Algorithm) generates the list of the NALs to request and from which neighbors.
- c. Parallel to these two main functions, the *BufferMap manager* and the *Description manager* acquire information about the data availability in the neighborhood (i.e. neighbors' BufferMaps) and the bitstream description (i.e. StreamMap). The *BufferMap manager* periodically asks the neighboring peers for their buffer maps in order to be aware of the available data chunks in the neighborhood. Similarly, the *Description Manager* asks periodically the neighboring peers for stream description portions.

V. PERFORMANCE EVALUATION

In this section, we evaluate the proposed solution using simulations.

A. Simulation Setup

To evaluate the performance of our preferential attachment algorithm, we measure the layer viewed by each peer during its session and the bandwidth utilization in order to show that the overlay formation guarantees better quality to each peer, and it is efficiently contributing to the overlay. We compare our proposed technique referred to in the following as *AFF (Affinity-based overlay formation)* against three other techniques as follows:

- a. *Random overlay formation (RND):* In this technique, a joining peer is connected randomly to M present peers.
- b. *Scale-Free overlay formation (SFR)* [41]: in this technique, a joining peer is connected to the M best serving peers. The serving capability (SC_i) of a peer is defined as the ratio of its uplink capacity to its out-degree.

$$SC_i = \frac{U_i}{O_i(t)}$$

- c. *Classified overlay formation (CLS)* [29]: In this technique, peers are organized into different clusters according to both download and uplink bandwidth. In *CLS* a peer can subscribe to different clusters in order to have a specified quality level.

For each evaluated techniques, we measure the following performance metrics:

- a. *Viewed Layer Distribution (VLD):* This metric provides an accurate view of the viewed layer by all the peers. It shows the proportion of peers that receive each layer of the video.
- b. *Download Bandwidth Utilization (DBU):* This metric measures the effective usage of the downlink capacity during the peer session.
- c. *Upload Bandwidth Utilization (UBU):* Similar to DBU, this metric measures the usage of the uplink capacity.
- d. *Overlay graph properties:* These include the average in-degree and out-degree of the peers.

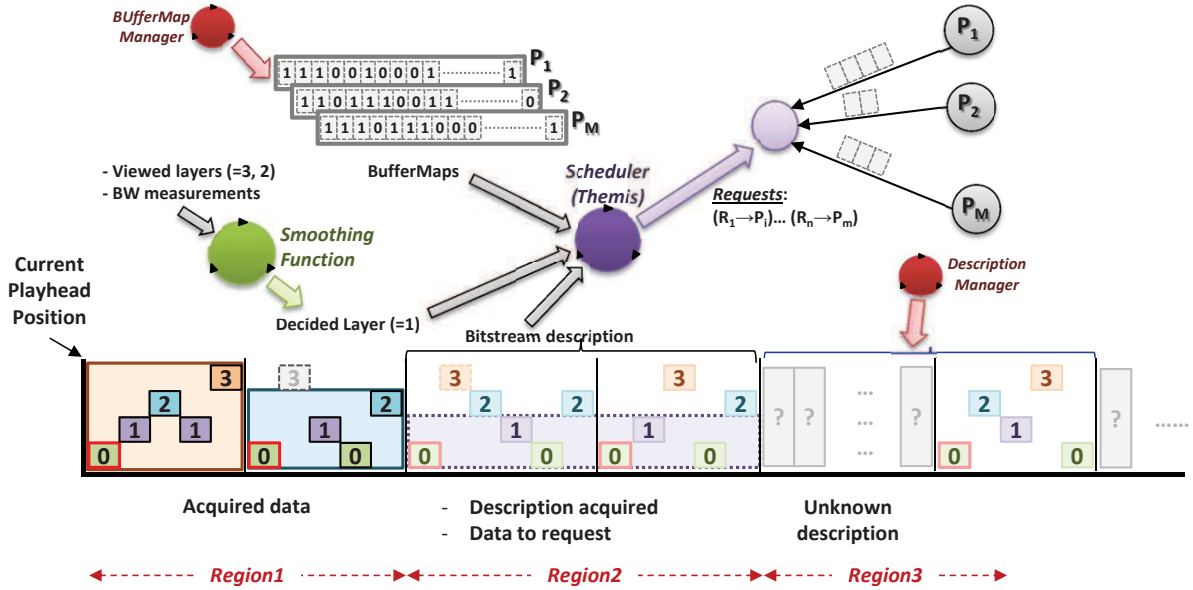


Fig. 10. Overview of the Streaming System

TABLE I
SIMULATION PARAMETERS

| Parameter | Value |
|----------------------|--|
| Peer lifetime | $\text{Expo}(\lambda = 1/300s^{-1})$ |
| Peer arrival | $\text{Expo}(\mu = 1/3s^{-1})$ |
| Simulation duration | 5 hours |
| Video quality levels | 1 video: 5 Layers $\{L_0, L_1, L_2, L_3, L_4\}$ Overall bitrate: 3000 Kbps $\{L_0\} = 1000$ Kbps $\{L_0, L_1\} = 1800$ Kbps $\{L_0, L_1, L_2\} = 2400$ Kbps $\{L_0, L_1, L_2, L_3\} = 2800$ Kbps $\{L_0, L_1, L_2, L_3, L_4\} = 3000$ Kbps |
| Download bandwidth | $\{1.0\text{Mbps}, 1.8\text{Mbps}, 2.4\text{Mbps}, 2.8\text{Mbps}, 3.0\text{Mbps}\} = \{20\%, 20\%, 20\%, 20\%, 20\%\}$ |
| Upload bandwidth | $\{1.20\text{Mbps}, 2.16\text{Mbps}, 2.88\text{Mbps}, 3.36\text{Mbps}, 3.60\text{Mbps}\} = \{20\%, 20\%, 20\%, 20\%, 20\%\}$ |

In our simulations, we have considered 5 different classes of peers having different network capabilities profiles. Each profile corresponds to a certain SVC layer that the peer can get according to its download capability. Peers are uniformly distributed among these 5 classes (20% each). Each class corresponds to a particular quality level. This means that the overlay is composed of 20% peers able to view video stream of quality 1 (i.e. L_0), 20% other peers are able to view video stream of quality 2 (i.e. L_0, L_1), etc. Upload and download capacities are set in order to have a network in surplus mode [42][43][44]. Table I summarizes the simulation parameters. We have used OMNeT++ [45], a discrete event simulator.

B. Simulation Results

The obtained results are shown in figures 11–15.

Viewed Layer Distribution: Figure 11 shows the distribution of the viewed layer by the peers. We can clearly see that the problem of quality bottleneck persists in the case of

the RND technique in such a way that 80% of the peers are being offered the Layer 0 only, and higher layers are only viewed by fewer peers (15%, 3%, 1%, 0% respectively). However, this problem is reduced when other techniques are used. Also, we can notice that these techniques allow a wider distribution of higher layers but still cannot distribute the highest layers (here layers 3 and 4) even if the overlay is composed of peers capable of receiving such layers. Finally, AFF technique performs better than SFR and CLS techniques. This can be explained by the fact that AFF technique benefits from both advantages of CLS and SFR techniques. The CLS technique allows the swarming of peers with similar network capabilities; whereas the load balancing feature of SFR technique prevents the formation of hotspots in the overlay (i.e. joining peers are attracted by peers with high upload capacity).

Bandwidth Utilization: Figures 12 and 13 show the bandwidth utilization in terms of the average download and upload rates during the session respectively. They also show the difference between the actual download and upload rates and their corresponding capacities. It is clear that in the RND-based solution, the overlay peers are less efficiently connected among them and this leads to poor download and upload rates. Therefore, a great portion of the available bandwidth (both upload and download) is unused. However, SFR and AFF techniques ensure better bandwidth utilization since the links' capacities are taken into account when connecting joining peers to the overlay.

Overlay Graph Properties: Figure 14 shows the input and output degrees of peers within the constructed overlay. We can see that SFR and AFF exhibit a small number of connections compared to RND and CLS. We can also see that SFR provides slightly slower degrees than AFF. Indeed, SFR only relies on the peer contribution (the ratio of the upload capacity to the out-degree of the peer). This technique avoids attaching too many peers to a node than it can handle with respect to its

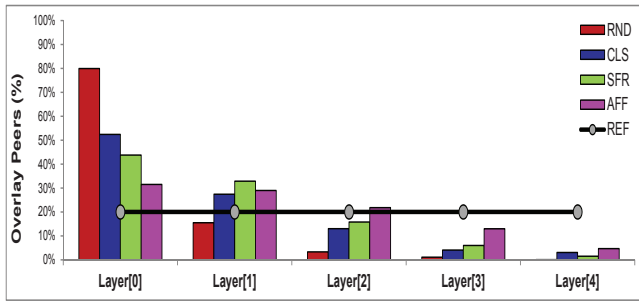


Fig. 11. Viewed Layer Distribution

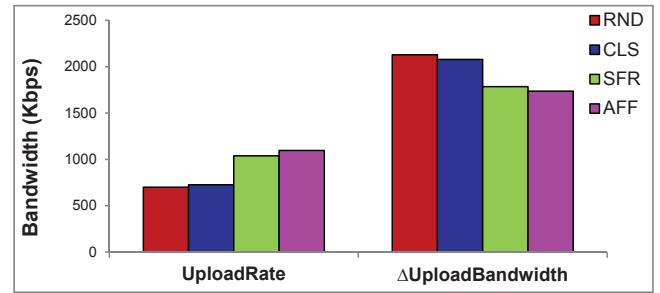


Fig. 13. Upload Bandwidth Utilization

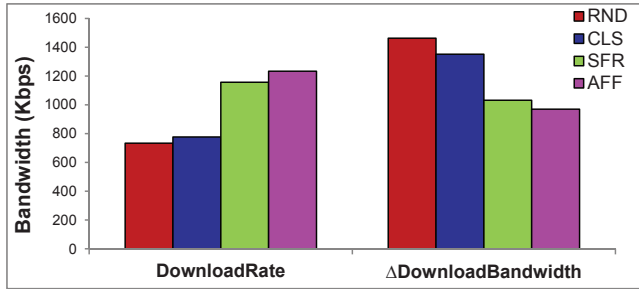


Fig. 12. Download Bandwidth Utilization

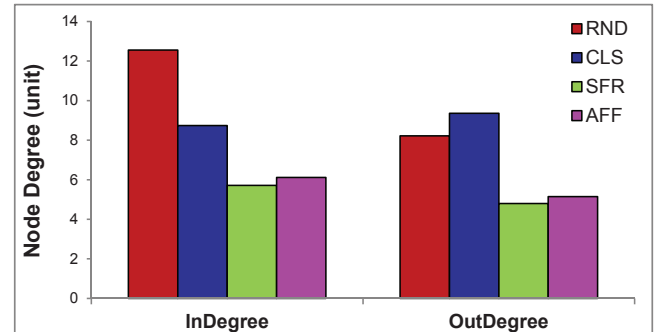


Fig. 14. Overlay Graph Properties

uplink capacity. On the other hand, AFF includes a similar rule but also considers the video quality level that can be provided by the sender peer and the quality level that can be obtained by a receiver peer.

Finally, the lower nodes' degrees demonstrate the efficiency of our proposed overlay formation technique since with fewer but more appropriate connections, it achieves better layer distribution and therefore avoids the quality bottleneck problem.

In order to study the impact of neighbors' churn on the streaming stability, we evaluated the Themis algorithm under different neighborhood configurations and measured the quality stability in terms of layer switching before the replacement of the departing neighbor. Figure 15 shows the distribution of the number of dropped layers following a neighbor departure. We can see that the Themis algorithm provides superior results in terms of quality stability. Thanks to the load balancing feature, the received quality level does not suffer at all in 28% of the time. Moreover, the Themis algorithm still provides good results in 65% of the time, since the number of dropped layers is 1 or less.

VI. CONCLUSION

In this paper, we proposed a comprehensive solution for adaptive SVC streaming over overlay networks. The solution integrates a set of strategies jointly used to avoid the quality bottleneck problem while ensuring efficient SVC streaming. Specifically, an overlay formation strategy and a data scheduling strategy are combined. The overlay formation strategy provides a stable neighborhood for a newly joining peer. It is based on preferential attachment that takes into account the quality level the peer can get from its neighbors, and the quality level it is willing to serve to other peers. This neighborhood is chosen in such a way to allow both

good throughput and peer contribution. We also proposed a data scheduling algorithm (*Themis*) to improve the video quality that each peer receives from its neighborhood. *Themis* relies on requests dispatching among the neighbors according to their upload capacities and the set of layers they can provide in order to mitigate the impact of neighbor departure. Simulations have been conducted to evaluate the performance of the proposed solution. We evaluated different metrics including churn tolerance for the data scheduling mechanism, and the spreading of higher layers for the overlay formation mechanism. The evaluation results confirmed the superiority of our approach in effectively addressing the quality bottleneck problem and ensuring a wide spreading of higher layers across the overlay network. They also demonstrated how our scheduling algorithm can provide each peer with a relatively stable quality and hence an improved quality of experience.

REFERENCES

- [1] G. Kreitz and F. Niemela, "Spotify—large scale, low latency, p2p music-on-demand streaming," in *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on*, pp. 1–10, IEEE, 2010.
- [2] "Pplive, <http://www.pptv.com/>," 2013.
- [3] "Zattoo, <http://www.zattoo.com/>," 2013.
- [4] Z. Huang, C. Mei, L. E. Li, and T. Woo, "Cloudstream: Delivering high-quality streaming videos through a cloud-based svc proxy," in *INFOCOM, 2011 Proceedings IEEE*, pp. 201–205, IEEE, 2011.
- [5] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *INFOCOM, 2012 Proceedings IEEE*, pp. 1620–1628, IEEE, 2012.
- [6] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau, "Internet video delivery in youtube: From traffic measurements to quality of experience," in *Data Traffic Monitoring and Analysis*, pp. 264–301, Springer, 2013.
- [7] M. Mushtaq and T. Ahmed, "P2p-based mobile iptv: Challenges and opportunities," in *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on*, pp. 975–980, IEEE, 2008.

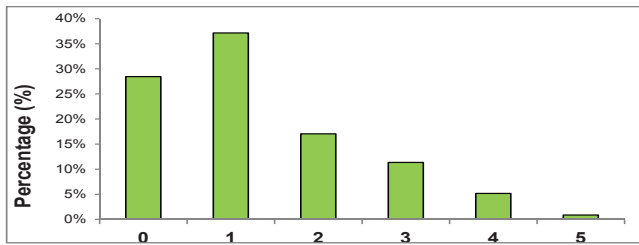


Fig. 15. Number of dropped layers in case of 1-neighbor departure

- [8] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, 2001.
- [9] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, 2007.
- [10] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *Network Protocols, 2003. Proc. 11th IEEE International Conference on*, pp. 16–27, IEEE, 2003.
- [11] Z. Liu, Y. Shen, K. W. Ross, S. S. Panwar, and Y. Wang, "Layerp2p: Using layered video chunks in p2p live streaming," *Multimedia, IEEE Transactions on*, vol. 11, no. 7, pp. 1340–1352, 2009.
- [12] J. Zhang, W. Xing, Y. Wang, and D. Lu, "Modeling and performance analysis of pull-based live streaming schemes in peer-to-peer network," *arXiv preprint arXiv:1302.4030*, 2013.
- [13] L. Han and H. P. In, "An adaptive scheduling algorithm for scalable peer-to-peer streaming," in *Computers, Networks, Systems, and Industrial Engineering 2011*, pp. 193–200, Springer, 2011.
- [14] X. Xiao, Y. Shi, Y. Gao, and Q. Zhang, "Layerp2p: A new data scheduling approach for layered streaming in heterogeneous networks," in *INFOCOM 2009, IEEE*, pp. 603–611, IEEE, 2009.
- [15] C. Liang, Y. Guo, and Y. Liu, "Is random scheduling sufficient in p2p video streaming?," in *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on*, pp. 53–60, IEEE, 2008.
- [16] A. Bradai, U. Abbasi, R. Landa, and T. Ahmed, "An efficient playout smoothing mechanism for layered streaming in p2p networks," *Peer-to-Peer Networking and Applications*, pp. 1–17, 2012.
- [17] U. Abbasi and T. Ahmed, "Smooth delivery of layered video stream in p2p networks," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 1986–1990, IEEE, 2012.
- [18] M. Moshref, R. Motamedi, H. R. Rabiee, and M. Khansari, "Layeredcast: A hybrid peer-to-peer live layered video streaming protocol," in *Telecommunications (IST), 2010 5th International Symposium on*, pp. 663–668, IEEE, 2010.
- [19] Y.-H. Moon, J.-N. Kim, and C.-H. Youn, "Churn-aware optimal layer scheduling scheme for scalable video distribution in super-peer overlay networks," *The Journal of Supercomputing*, pp. 1–21, 2013.
- [20] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: Optimal performance trade-offs," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, pp. 325–336, ACM, 2008.
- [21] P. Liu, G. Huang, S. Feng, and J. Fan, "Event-driven high-priority first data scheduling scheme for p2p vod streaming," *The Computer Journal*, vol. 56, no. 2, pp. 239–257, 2013.
- [22] M. Gaggio, S. Milani, L. Lima, P. Migliorati, and G. Calvagno, "P2p scalable video streaming using data priority and fec-based noncooperative multiple description," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pp. 1534–1538, IEEE, 2012.
- [23] H. Zhou, Y. Ji, Y. Gu, and B. Zhao, "Network coding based svc multicast over broadband wireless networks," in *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*, pp. 428–435, IEEE, 2012.
- [24] A. T. Nguyen, B. Li, and F. Eliassen, "Chameleon: Adaptive peer-to-peer streaming with network coding," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, 2010.
- [25] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," in *ACM SIGOPS Operating Systems Review*, vol. 40, pp. 333–344, ACM, 2006.
- [26] F. Dobrian, A. Awan, D. Joseph, A. Ganjam, J. Zhan, V. Sekar, I. Stoica, and H. Zhang, "Understanding the impact of video quality on user engagement," *Communications of the ACM*, vol. 56, no. 3, pp. 91–99, 2013.
- [27] Y.-F. Ou, H. Zeng, and Y. Wang, "Perceptual quality of video with quantization variation: A subjective study and analytical modeling," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pp. 1505–1508, IEEE, 2012.
- [28] R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang, "Qdash: A qoe-aware dash system," in *Proc. 3rd Multimedia Systems Conference*, pp. 11–22, ACM, 2012.
- [29] G. Zhang and C. Yuan, "A classified p2p overlay scheme using svc for video streaming," in *Advances in Multimedia Information Processing-PCM 2009*, pp. 440–450, Springer, 2009.
- [30] M. Fesci-Sayit, E. T. Tunali, and A. M. Tekalp, "Bandwidth-aware multiple multicast tree formation for p2p scalable video streaming using hierarchical clusters," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 945–948, IEEE, 2009.
- [31] A. Zambelli, "Microsoft iis smooth streaming technical overview," *Microsoft Corporation*, vol. 3, 2009.
- [32] "Adobe, using adobe http dynamic streaming," 2010.
- [33] "Apple, http live streaming overview," 2009.
- [34] R. Pantos, "Http live streaming," 2012.
- [35] N. Zong, "Survey and gap analysis for http streaming standards and implementations," 2010.
- [36] S. Medjiah, T. Ahmed, E. Mykonati, and D. Griffin, "Scalable video streaming over p2p networks: A matter of harmony?," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2011 IEEE 16th International Workshop on*, pp. 127–132, IEEE, 2011.
- [37] M. Eberhard, A. Palo, A. Kumar, R. Petrocco, L. Mapelli, and M. Uitto, "Nextsharepc: an open-source bittorrent-based p2p client supporting svc," in *Proc. 3rd Multimedia Systems Conference*, pp. 101–106, ACM, 2012.
- [38] T. Silverston and O. Fourmaux, "Measuring p2p iptv systems," in *Proc. of ACM NOSSDAV*, vol. 7, 2007.
- [39] D. Leonard, Z. Yao, V. Rai, and D. Loguinov, "On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 644–656, 2007.
- [40] C. Wu, B. Li, and S. Zhao, "Characterizing peer-to-peer streaming flows," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1612–1626, 2007.
- [41] A.-L. Barabási, "Scale-free networks: A decade and beyond," *Science*, vol. 325, no. 5939, pp. 412–413, 2009.
- [42] H. Luan, K.-W. Kwong, Z. Huang, and D. H. Tsang, "P2p live streaming towards best video quality," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pp. 458–463, IEEE, 2008.
- [43] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for p2p streaming systems," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 919–927, IEEE, 2007.
- [44] C. Huang, J. Li, and K. W. Ross, "Can internet video-on-demand be profitable?," in *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 133–144, ACM, 2007.
- [45] A. Varga *et al.*, "The omnet++ discrete event simulation system," in *Proc. European Simulation Multiconference (ESM2001)*, vol. 9, p. 185, sn, 2001.



Samir Medjiah Dr. Samir Medjiah is a Teaching and Research Assistant (ATER) at IPB (Institut Polytechnique de Bordeaux) in the ENSEIRB-MATMECA school of engineering. He is doing his research activities in CNRS-LaBRI Lab, UMR 5800 at Univ. de Bordeaux. Samir Medjiah received his M.S in Computer Science with distinction from National Institute of Computer Science (ESI ex-INI), Algiers, Algeria, in 2009, and his PhD with high honors from LaBRI/University of Bordeaux-1, France, in 2012. His main research

interests are multimedia communications over challenged networks including routing, transport and congestion control of multimedia streams in wireless multimedia sensor networks, delay-tolerant networks, satellite and space communications, and Internet overlay networks. He has won national competitions in the field of mathematics (Mathematical Olympiad 2004) and computer science. Dr. Medjiah is an IEEE/ComSoc member.



Toufik Ahmed Dr. Toufik Ahmed is a Professor at IPB (Institut Polytechnique de Bordeaux) in the ENSEIRB-MATMECA school of engineering. He is doing his research activities in CNRS LaBRI Lab, UMR 5800 at University of Bordeaux 1. Toufik Ahmed received the B.Sc. in Computer Engineering, with high honors, from the, the I.N.I (National Institute of Computer Science) Algiers, Algeria, in 1999, an M.Sc. and a Ph.D degrees in Computer Science from the University of Versailles, France, in 2000 and 2003, respectively. In November 2008, he

obtained the HDR degree (Habilitation Diriger des Recherches) at University of Bordeaux -1 on Adaptive Streaming and Control of Video Quality of Service over Wired / Wireless IP Networks and P2P Architectures. Toufik Ahmed was a visiting scientist at the School of Computer Science of the University of Waterloo in 2002 and research fellow at PRISM laboratory of the University of Versailles until 2004. His main research activities concern Quality of Service for multimedia wired and wireless networks, end-to-end signaling protocols, P2P network and Wireless Sensors Network. His work on quality of service and video delivering has led to many publications in major journals and conferences.



Rouf Boutaba Dr. Raouf Boutaba received the M.Sc. and Ph.D. degrees in computer science from the University of Pierre and Marie Curie - Paris 6, France, in 1990 and 1994, respectively. He is currently a Full Professor of computer science at the University of Waterloo, Waterloo, ON, Canada. His research interests include network, resource and service management in wired and wireless networks. He is the founding editor in chief of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT (2007-2010) and on the

editorial boards of several other journals. He has received several best paper awards and other recognitions such as the Premiers Research Excellence Award, the IEEE Hal Sobol Award in 2007, the Fred W. Ellersick Prize in 2008, the Joe LociCero and the Dan Stokesbury awards in 2009, and the Salah Aidarous Award in 2012. He is a fellow of the IEEE and the Engineering Institute of Canada.