On the Impact of Quality Adaptation in SVC-based P2P Video-on-Demand Systems

Osama Abboud[*]
Multimedia Communications Lab, TU Darmstadt, Germany
abboud@kom.tu-darmstadt.de

Thomas Zinner
Chair of Communication Networks, University of Wuerzburg, Germany
zinner@informatik.uni-wuerzburg.de

Konstantin Pussep
Multimedia Communications Lab, TU Darmstadt, Germany
pussep@kom.tu-darmstadt.de

Sabah Al-Sabea[†]
Internet Services Group
NEC Europe Ltd.
Heidelberg, Germany
al-sabea@neclab.eu

Ralf Steinmetz
Multimedia Communications Lab, TU Darmstadt, Germany
steinmetz@kom.tu-darmstadt.de

ABSTRACT

P2P Video-on-Demand (VoD) based on Scalable Video Coding (SVC) (the scalable extension of the H.264/AVC standard) is gaining momentum in the research community, as it provides elegant adaptation to heterogeneous resources and network dynamics. The major question is, how do the adaptation algorithms and designs affect the overall perceived performance of the system? Better yet, how can the performance of an SVC-based VoD system be defined? This paper explores the impact and trade-offs of SVC-based quality adaptation with focus on the SVC layer selection algorithms, which are performed at different streaming stages. We carry out extensive experiments to evaluate the performance in terms of session quality (start-up delay, video stalls) and delivered SVC video quality (layer switches, received layers), and find out that these two metrics exhibit a trade-off. Our analysis and conclusions give multimedia providers insights on how to design and fine-tune their VoD system in order to achieve best performance.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Algorithms, Design, Performance

Keywords

Peer-to-Peer, Overlay Networks, Scalable Video Coding, Quality Adaptation, Video-on-Demand, Content Distribution.

1. INTRODUCTION

Demand for multimedia applications has witnessed enormous growth recently [9]. To catch-up with this growth, various service architectures ranging from client/server to distributed cloud approaches have been developed. One promising architecture is the peer-assisted delivery, which relies on a delicate balance between client/server and Peer-to-Peer (P2P) delivery techniques. Peer-assisted[1] Video-on-Demand (VoD) systems utilize the idea of peers assisting the servers by uploading chunks they have already downloaded. Thereby, peer-assisted VoD allows for either a higher number of supported peers, or a higher bit-rate for the same number of peer and server resources.

Nonetheless, peer-assisted VoD is still challenging. In current systems, video bit-rates typically range from 300 Kbps to 2 Mbps [11]. While lower bit-rates are still preferred by the content providers in order to support a wider spectrum of end-user devices, higher bit-rates are increasingly getting popular. Additionally, there is a need to support the heterogeneity of Internet devices, e.g. PCs, tablet computers, and mobile devices, within the same video delivery system.

Prominent approaches to achieve this goal are: Multiple Descriptor Coding (MDC) and Scalable Video Coding (SVC). In this paper, we focus on the latter as it exhibits a lower complexity, and allows for adaptation to resources with three degrees of freedom. Receivers can have different screen resolutions, heterogenous link capacities, and different processing capabilities [1].

Combination of P2P VoD and SVC with full-fledged adaptation features raises many challenges and questions: Does SVC really help in systems with heterogenous resources?

[*]This work was funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (Support code 01 BK 0806, GLab). The authors alone are responsible for the content of the paper.

[†]Work done before joining NEC Europe Ltd.

MMSys'11, February 23–25, 2011, San Jose, California, USA.
Copyright 2011 ACM 978-1-4503-0517-4/11/02 ...$10.00.

[1] We use the terms peer-assisted VoD and P2P VoD interchangeably in this paper.

How should the layer selection algorithm choose the layer to be fetched? How often should these algorithms be executed? What is the impact these adaptation algorithms have on the perceived system quality? And also, how to measure the quality of the VoD system?

These questions are not only posed by current developments of VoD applications [3], but also by content providers that have to embrace adaptation techniques for delivering multimedia to devices with a diverging resource spectrum.

In this paper, we address these questions using extensive simulations. We not only present performance metrics that assess the tradeoffs of an SVC-based VoD system, but also investigate the impact of adaptation and how it can be tweaked to reach best performance. We base our analysis on the VoD system presented in [1], while further elaborating the architecture and algorithms behind our VoD system.

This paper is structured as follows, Section 2 provides related work. Background on SVC is given in Section 3. Our proposed architecture for P2P VoD is presented in Section 4. Section 5 gives an overview on our used methodology. In Section 6, we present our simulative analysis and then conclude the paper in Section 7.

2. RELATED WORK

The research community has put substantial effort into investigating P2P VoD [16]. This includes different aspects such as prefetching policies [12], theoretical models [14], replication techniques [6], network-awareness [10] and impact of server allocation [23]. It was early recognized that prefetching and coding techniques are crucial for a high streaming experience [4], while Chi *et al.* proposed to combine network coding techniques with deadline awareness [7].

However, in scenarios with heterogeneous user devices, media coding techniques such as SVC allow to operate in the presence of devices with varying resources, from desktop computers to handhelds [8]. Furthermore, quality can be switched during playback to adapt to changing network conditions and system load. To support a wide spectrum of resources within the same system, various adaptation techniques were considered [5, 17, 20, 22, 25]. For example, PALS [25] is a receiver driven P2P video streaming system with quality-adaptive playback of layered video. However, PALS only considers single dimensional scalability (as the case for many layered streaming systems) and, therefore, cannot adapt to heterogeneous characteristics of peers with different degrees of freedom.

Different live video streaming approaches have been developed and discussed in research in recent years. Baccichet *et al.* [5] use prioritization of packets and multicast trees to distribute SVC sub-streams with a bound on the introduced delay. Lee *et al.* focus in [15] on challenges for segment seeding and scheduling while deploying a live P2P streaming system using SVC. In [20], Nguyen *et al.* present and analyze a streaming system designed to incorporate network coding and SVC to facilitate deployment of adaptation techniques in streaming systems.

In contrast to the mentioned pieces of work, we present our P2P VoD system with focus on the impact of adaptation and layer selection algorithms. Therefore, we use three dimensional scalability as defined by the H.264/SVC standard [26] to adapt to different peer resources and network conditions.

Regarding P2P-based VoD systems that combine P2P and SVC, several architectures have been recently proposed. For instance, [19] proposes and evaluates a system that aims at achieving quality adaptation and a smooth playback. However, the authors do not investigate a real P2P system, but rather focus on a local view (simple download of content from several peers). Mokhtarian *et. al* present in [18] an analysis of peer-assisted VoD systems with scalable video streams. They provide analytical models that estimate the number of peers that can be admitted into the system in case of flash crowds. Their results can be integrated into our analysis to better match server and peer resources.

Another approach, introduced by Oechsner *et. al* [21] is similar to ours. However, the authors investigate only temporal scalability and evaluate the mean played-out layers and stalling times. This differs from our approach, since we additionally investigate the trade-off between SVC video quality and session quality. We also investigate performance metrics per peer. This is of special interest, since then we can better assess the impact of adaptation on the system as a whole as well as on individual peers.

SVC allows adaptation to be performed at the receiving peers by simply requesting parts of the stream that match their resources. Nonetheless, the existence of more powerful peers enables approaches where those peers can actively reencode the video file to match resources of the receiving peers as done in [13].

3. SCALABLE VIDEO CODING

The video codec H.264/SVC (Scalable Video Coding) [26] is based on the H.264/AVC (Advanced Video Coding) standard, a video codec widely used in the Internet, for instance by video platforms (e.g., YouTube, GoogleVideo) and video streaming applications (e.g., Zattoo). H.264/AVC is a single-layer codec, which means that different copies of the same video streams have to be encoded to support different end-user devices. Using the scalable video coding extension of H.264/AVC, a video file can be encoded at different qualities within the same layered bitstream. This includes different resolutions, different frame-rates and different picture qualities with respect to the Signal-to-Noise Ratio (SNR). These three dimensions of scalability are denoted by spatial, temporal and quality respectively.

Figure 1 gives an example on the different possible scal-

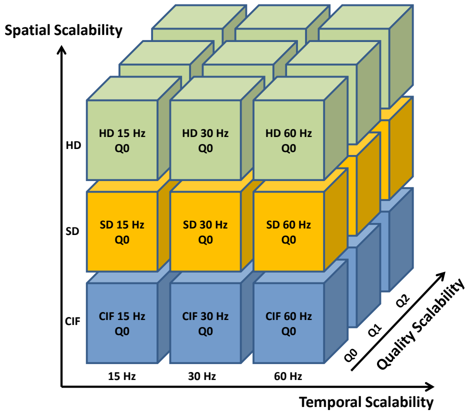


Figure 1: SVC cube model, illustrating the possible scalability dimensions in an SVC video file

abilities offered by an SVC file. This scalable video can be viewed in three different spatial resolutions (CIF, SD, HD), three different temporal resolutions (15Hz, 30Hz, 60Hz), and three different quality resolutions (Q0, Q1, Q2). The left bottom sub-cube, CIF resolution with 15 Hz and quality Q0, is called the base layer and is essential for playback. Based on this base layer, different enhancement layers permit a better video quality with a higher resolution, higher frame rate and/or better SNR. The more sub-cubes along any of the three axes are included, the higher the quality is in the respective dimension. If all sub-cubes are available, the video can be played back with highest quality. The possibility to switch seamlessly between different qualities enables an adaptation of the video quality to device resources and capacity of the system. Thus, it is possible to skip some enhancement layers in case of insufficient resources while enabling a continuous playback of the video.

To allow for multi-source streaming, an SVC file is divided into many chunks, which can be played independently. Usually, a chunk is worth 0.5 up to 2 seconds of video playback. Each chunk can be streamed with different qualities following the SVC model. A chunk is further divided into multiple blocks each contributing to a different quality level. A block will be used as basic unit for fetching and distributing video data across the network.

4. THE QUALITY-ADAPTIVE SYSTEM

In this section, we present our SVC-based quality-adaptive VoD system. We assume a mesh-based pull approach for VoD as presented in [1]. We further assume having a tracker that keeps track of all peers in the network. To ensure a certain quality of service, servers with modest resources are deployed, which additionally inject the initial content. Figure 2 depicts the basic architecture of our quality adaptation workflow.

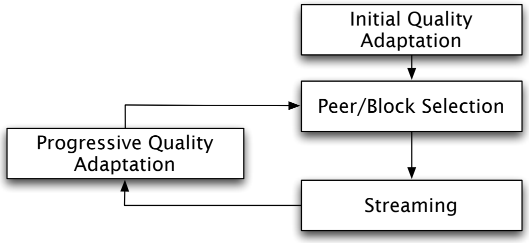


Figure 2: The quality adaptation workflow

Quality adaptation is achieved by adjusting quality according to the different peer resources and network dynamics. It is performed by two modules: the Initial Quality Adaptation (IQA) and the Progressive Quality Adaptation (PQA). Both modules form the algorithms that match the layers with resources available at the peer. On the one hand, the IQA is used for determining the highest possible layer that a peer can retrieve and play, and is performed at session start. On the other hand, the PQA is performed periodically to adjust the layer according to the changes of the network environment.

After the playback is initiated, the IQA is first called to make a decision on the feasible quality level based on local resources. Based on this decision, peer selection and block selection are performed. Peers are selected in such a way that they are able to provide the selected layer. After the neighboring peers have been contacted and upload slots have been reserved, block selection is done. To ensure continuous playback, the PQA is performed regularly, and if required, it may increase or decrease the selected layer accordingly.

Next, we give the details of the quality adaptation modules and their role in the VoD system.

4.1 Initial Quality Adaptation

Initial Quality Adaptation (IQA) is typically invoked only once at the beginning of the playback session. It is designed in such a way that each peer can determine its highest SVC quality level before starting to download the SVC video. The architecture of the IQA is depicted in Figure 3.

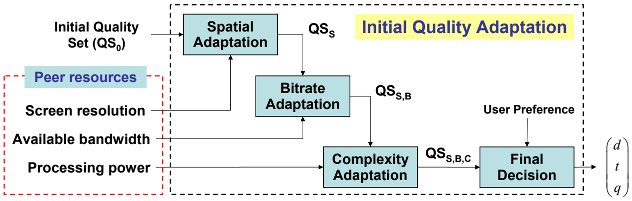


Figure 3: Initial quality adaptation [1]

The basic idea of IQA is to compare the requirements of each layer of the video stream with the local resources[2] of a peer so that the layers that are not supported are left out of the decision process. The subtle property of the IQA, is that it has to make a decision on the quality level without having any information about system throughput and dynamics.

According to the three dimensions of scalability in the SVC model, we have identified the following relevant local resources of a peer:

- *Screen resolution.* The resolution of the user's display determines the picture size of the video to be downloaded. Therefore, the screen size will be used to restrict the spatial quality-level of a video.
- *Bandwidth.* The download bandwidth of a peer corresponds to the maximal bit-rate of the video stream it can receive. Therefore, bandwidth sets limits on the bit-rates of the streamable SVC layers.
- *Processing power.* Decoding SVC streams requires more processing power compared to non-scalable streams. Additionally, the more processing power a peer has, the more layers it can decode. Therefore, it is necessary to take the available processing power into consideration, which would set limits on the decoding complexity of the SVC layers.

Using the three types of local resources, we can filter out those SVC layers that are not compatible by cross-checking them with local resources. The IQA starts off from an initial quality set QS_0 that contains all possible layer combinations. This set is passed through the spatial, bit-rate, and complexity adaptation sub-modules to generate the lists QS_S, $QS_{S,B}$, and $QS_{S,B,C}$ respectively. Each sub-module filters out all incompatible layers. For example, if the bandwidth of a certain peer is BW, then all layers with bit-rates larger

[2] Local resources here mean hardware resources that do not change dynamically.

than BW are filtered out. The same is also done for the screen resolution and processing power.

Final Decision. The *final decision* algorithm continuously receives a list of feasible layers from the previous adaptation algorithms. It is then up to this module to make a final decision on the layer to be fetched in the next round. The straight-forward decision is to choose the highest layer in order to give the user the best possible video quality. However, it is sometimes difficult to choose the best one from a number of SVC layers. For example, given a $2 \times 2 \times 3$ SVC video file, we could have the layer $(1, 1, 2)$ (higher SNR quality) and $(1, 2, 1)$ (higher frame rate) as candidates for the best layer after the IQA or PQA. Here one cannot tell which layer is better. In this case, the final decision of which one to choose depends on the user's preference (for example, configured by the user in the application settings).

4.2 Progressive Quality Adaptation

The Progressive Quality Adaptation (PQA) module is the dynamic part of the quality-adaptive system. The PQA is invoked regularly during the video playback, with configurable time intervals to adjust the SVC layer according to the different system dynamics. Thus, potential stalls can be avoided and a smooth playback is ensured. The architecture of the PQA is depicted in Figure 4.

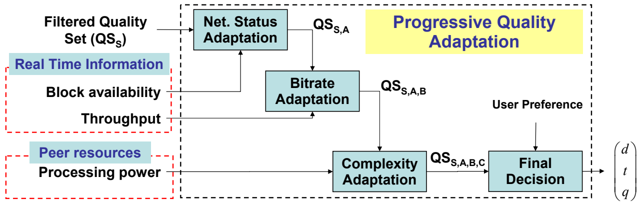


Figure 4: Progressive quality adaptation [1]

In addition to using complexity adaptation, as for the IQA, the PQA uses real-time information of network status measured through the *block availability* in the neighborhood and the active *download throughput*. It takes the current SVC layer as input, then adjusts it according to the real-time network information. This layer is processed by the different stages of the PQA to produce a new layer that fits the current network conditions.

The three adaptation stages of the PQA form together the decision-making process. Since the screen size of the user does not change during the video playback, the PQA adjusts only the temporal and SNR dimensions. The spatial layer will not be changed by the PQA.

We now give more details about the three PQA adaptation algorithms.

Net-status Adaptation. This part of the PQA keeps track of the block availability of all connected peers. Its objective is to check whether the current layer can be supported by the available blocks from current neighbors. Here ”support” means that all the blocks in the high priority or buffering window can be downloaded without changing the currently connected peers. If this cannot be guaranteed, the SVC layer of the local peer will be decreased to avoid performance degradation until new peers have been contacted. The adaptation process can be briefly described in the following steps:

1. The local peer uses the information of all available blocks at its connected peers acquired through the so-called buffer-maps.
2. Then, the local peer can calculate a neighborhood availability map for the blocks in its high priority window.
3. The availability map is then compared with the current layer of the local peer. If the map covers all blocks to be downloaded for this layer in the high priority window, the current layer will not be changed. Otherwise, the layer will drop to the level that is covered by the availability map. In addition, if the availability map contains additional blocks of a higher layer, the PQA then switches to this layer.

Using PQA, the SVC layer of a peer can be adapted according to the real-time resources of its connected peers, so that the playback does not need to stop and wait for unavailable blocks. Consequently, the number of stalls can be reduced during the playback.

Bit-rate Adaptation. This stage of PQA affects the SVC layer by analyzing the change of download throughput during the buffering process. The goal of the bit-rate adaptation is to predict possible buffer-underflows due to slow block supply, then adapt the layer so that the bit-rate fits the throughput, therefore, avoiding potential stalls.

To realize this goal, we first have to answer this question: How can we efficiently measure a peer's download throughput of the high priority set of the current layer? To do this, we have observed the following: The buffering state, which means how full the buffer is, reflects the recent download throughput for the current layer. Based on this observation, we can measure the throughput by monitoring the buffer state. Therefore, a nearly full buffer indicates a high throughput, while an almost empty buffer indicates a low throughput. With the above observation, we realize throughput adaptation as follows:

1. We measure how much the buffer is full with video data for the current layer.
2. If this portion is very low (say, less than 10%), then we decrease by one the SNR or temporal level of the current layer.
3. If the state of the buffer is good (e.g. more than 80% is filled), and the current layer is below the initial level determined by the IQA, then we increase the current SNR or temporal layer by one level towards its IQA upper bound.

Complexity Adaptation. The complexity adaptation algorithm is responsible for checking whether the currently available processing power of the local peer is sufficient for decoding the selected SVC layer. Therefore, this part of the PQA would increase or decrease the selected layer according to the current processor load of the device.

Finally, if the different adaptation algorithms result in more than one layer possibility, the predefined user preference is again used to make the final decision.

4.3 Peer Selection

The quality level of each peer has to be taken into consideration during peer discovery and selection. Since each peer in the VoD system has its own SVC layer, not all peers registered at the tracker can support this quality level. Therefore, more information is needed at the tracker in order to match the layer offer and demand. Thereby, the *current layer* of the peer is included into the *peer-discovery request* sent to the tracker. The tracker can then return only those peers that can support the given quality level. The tracker further stores this quality level in its local database for further peer discovery requests. Later on, each peer would announce its current layer with each keep-alive message sent to the tracker to keep the information there as fresh as possible.

The idea behind the modified peer selection algorithm is to have neighborhood peers whose layer is possibly equal to or higher than that of the requesting peer. Therefore, it is more probable that any of the contacted peers can potentially provide any block needed by the downloading peer.

4.4 Connection Management

Algorithms for managing connections are divided into downloader and uploader side algorithms:

Downloader Side. Dowloader peers periodically send connection requests to other peers. If the reply is positive, then the downloader peer requests the blocks it needs, which are further transmitted. If the reply is negative, then the downloader peer degrades the rank of the remote peer to avoid keeping useless connections. Unused connections will be eventually replaced by new ones.

Uploader Side. After an uploader peer receives a new connection request, it checks whether it still has free slots to accept the request. If it does, then a slot is assigned and the request is served. If there are no free slots, the peer evaluates the urgency of the request. If the request is more urgent than one of the existing connections, then it assigns the new connection to the slot whose connection has the lowest priority. Then, this connection is dropped gracefully.

4.5 Block Selection

Block selection has the main role of assigning each block a certain priority. In addition, it is sometimes required to skip some blocks to allow for continuous playback. With the information of the current layer, a peer does not need to download every block of the original video stream, but only those that belong to the selected quality level.

We still follow the idea of dividing the remaining video data into two zones according to the priority: high priority video blocks and low priority video blocks [16]. The partition of these two priority zones is similar to that of a general video streaming system (namely using the buffering window as high-priority zone and the rest of the video as low-priority zone). To calculate the download priority for each block, we have to consider not only its position in the temporal domain, but also its quality level in the SVC model.

High-Priority Zone. The download-priority of each block in the high-priority zone is determined by considering two factors: its distance to the current playback position and its quality level in the SVC model. Therefore, a block n is assigned a certain priority as follows:

$$Priority(n) = -A\frac{n-P}{HP\_Size} - B(W_d d + W_t t + W_q q). \quad (1)$$

The left part of Equation 1 represents the temporal priority by taking the distance between the playback position (P) and the block number (n) into account. The right part of the equation generates the SVC priority: a value which sinks with an increased quality level in any dimension. With the coefficients W_t, W_d and W_q, the speed of the priority-drop in any dimension of the SVC model can be controlled. Finally, the two parts are added together with the weights A and B so that a balance between temporal urgency and SVC quality is ensured.

Low-Priority Zone. For the low-priority zone, we use an algorithm that favors prefetching blocks that will soon be needed by peers downloading from the local peer as presented in [2]. Prefetching is started once the high priority set is full. Therefore, only peers with excess resources would actually perform the following strategies.

The local peer sends a request to all the peers in its upload neighborhood querying for votes on their most wanted blocks. On receiving such a request, each peer places its votes starting from the first non-received block. Those votes are decreasing with increasing block number. The local peer then sums up the votes for each block and then sorts those blocks according to their vote values, i.e. importance. The block with the highest vote, which is not yet available at the local peer, will be prefetched and made available to the neighboring peers. The local peer filters out those blocks that do not belong to its IQA layer. The last step ensures that the selected blocks for prefetching should also be possibly playable by the peer itself, so that unnecessary downloads can be avoided. More details about this prefetching algorithm can be found in [2].

5. METHODOLOGY AND METRICS

Before we can evaluate the performance of the VoD system and the impact of quality adaptation, we first need to define relevant metrics that reflect the key features of perceived video quality. These metrics can be divided into two main categories: *session quality* and *SVC video quality* metrics.

5.1 Session Quality

In this category, we consider the most important factors that affect the users' watching experience in any VoD system. These are the *start-up delay* and *video stallings* that occur during the playback.

- *Start-up delay.* With this metric we measure how long the user waits until the playback begins. The shorter this time interval, the better is the session quality.
- *Stalling events per peer.* This metric reflects the frequency of stalling events taking place during the video playback, i.e. video freezes due to empty buffer. Therefore, in order to improve the session quality, the total number of stalling events should be minimized.
- *Average stalling duration.* In addition, we are also interested in the average duration of stalling events that happen during video playback. Again the shorter the stalling event, the better is the session quality.
- *Relative playback delay per peer.* For simplicity, sometimes we would like to have one metric that summarizes the above mentioned metrics. Therefore, we define the total playback delay as the sum of the start-up delay and all the stalling time. The relative delay is the total playback delay normalized by the total playback time, as shown in Equation 2.

$$Relative_{delay} = \frac{Delay_{init} + \sum_{i=1}^{n} Stall_i}{Time_{playback}} \quad (2)$$

5.2 SVC Video Quality

In addition to the session quality metrics, we are also interested in assessing the respective SVC video quality. This enables us to better judge the overall performance of the VoD system. Here the metrics of interest are:

- *Number of layer changes during video playback.* Some studies [27] have reported that having too frequent layer variations might be more annoying for users than watching the lowest quality. Therefore, we measure the average number of SVC layer variations as an indictor of SVC video quality. A smaller number of layer changes indicates a better VoD system.
- *Relative Received Layer.* In addition, the level of the SVC layer received by each peer during the playback is important. Since each peer has different local resources and thus can retrieve only a certain range of SVC layers, we cannot directly use the *absolute* layers received by the peers to compare their performance. Instead, we define the *relative layer* to assess whether the peers are receiving the highest quality they can actually get given their resources. The relative layer of each received video chunk is equal to the received layer divided by the initial SVC layer calculated by the IQA, as follows:

  $$Quality_{rel}(d,t,q) = \frac{d+t+q}{D_{init}+T_{init}+Q_{init}}, \quad (3)$$

  where d, t, q are the received layers in spatial, temporal and SNR-dimension respectively as chosen by the PQA, while $D_{init}, T_{init}, Q_{init}$ are the initial SVC layers as chosen by the IQA.
  Since the layer selected by the PQA can never be higher than that selected by the IQA (which is determined by physical resources), the relative received layer calculated by Equation 3 falls into the interval [0,1] for all peers. Using this metric, we can better compare the received SVC layer for peers with different local resources. A higher value of the relative received layer indicates that the peer is better able to maintain the quality supported by local resources. Having a lower value, on the other hand, means that although there are enough local resources, the P2P network itself is not able to provide the highest layer to the peer. This can be due to network congestions or to weak server resources.

6. EVALUATION

Here we present our simulative evaluation of the P2P VoD system. The goals of this study are: to assess the importance of quality adaptation using SVC, to measure the impact of adaptation, as well as to identify the tradeoffs of our system.

We focus mainly on three points: first, we want to see the impact of quality adaptation on the performance in comparison to a non-adaptive VoD system, i.e. a media agnostic one. Second, we are interested in the impact of the different quality adaptation algorithms. Finally, we are going to investigate how having different invocation intervals of the PQA affects the performance.

6.1 Scenario

Table 1 gives an overview on the used SVC video file. This model has 3 spatial levels (d), 4 temporal levels (t), and 1 SNR level (q), with a total of 12 SVC layer combinations. The rightmost column represents the total bit-rate of the respective quality level. The data in Table 1 was extracted from a real 5-minute SVC video file, which was encoded using the JSVM SVC Reference Software [24].

Parameter	Value
Simulation duration	200 minutes
Number of peers	90
Peer arrival pattern	Exponential
Number of servers	4
Server upload capacity	6 Mbps
Play-out buffer size	7 seconds
Neighborhood size	10
Video length	5 minutes

Table 2: Simulation setup

Our simulation setup is depicted in Table 2. We run simulations for 200 minutes during which 90 peers arrive based on an exponential distribution. To ensure a certain quality of service, we consider having 4 servers each with 1Mbps upload capacity. The playout buffer was chosen to be 7 seconds to ensure a small startup time and acceptable playback delay. The peers maintain a neighborhood of 10 peers.

Peer resources are configured as shown in Table 3. The

SVC layer (d,t,q)	Picture size	Frame rate (fps)	Partial Bit-rate (Kbps)	Total Bit-rate (Kbps)
0,0,0	176 × 144	3.75	60	60
0,1,0	176 × 144	7.5	30	90
0,2,0	176 × 144	15	30	120
0,3,0	176 × 144	30	30	150
1,0,0	352 × 288	3.75	180	240
1,1,0	352 × 288	7.5	90	330
1,2,0	352 × 288	15	60	390
1,3,0	352 × 288	30	60	450
2,0,0	352 × 288	3.75	270	510
2,1,0	704 × 576	7.5	150	660
2,2,0	704 × 576	15	180	840
2,3,0	704 × 576	30	160	1000

Table 1: SVC video structure with respective quality levels, partial bit-rates, and total bit-rates

	Set 1	Set 2	Set 3
Number	30	30	30
Screen size	176 × 144	352 × 288	704 × 576
Upload speed	128 Kbps	320 Kbps	800 Kbps
Download speed	256 Kbps	560 Kbps	1200 Kbps

Table 3: Resource configuration for the peers

given values help us to assess the impact of heterogenous resources in terms of bandwidth and screen sizes. Therefore, there are 3 groups of peers each with different bandwidth and local resources.

6.2 Results and Analysis

6.2.1 *Quality Adaptative Versus Media Agnostic VoD*

Now we evaluate how our proposed adaptation algorithms improve the performance of the P2P VoD system. We simulate our streaming system in three different cases: with no adaptation at all i.e. all peers try to retrieve the highest layer possible as in any media agnostic system, with adaptation algorithms utilizing first only IQA and then with both IQA and PQA.

The results are presented in Figure 5. The left sub-figure illustrates the average number of stalls while the right subfigure illustrates the total playback delay for each peer in the network. The x-axis refers to the peer IDs in the VoD network. For better comparability, we present the per-peer results in an increasing order and further divide the results into three groups according to the bandwidth capacities of the peers, starting from slowest (on the left) to the fastest (on the right). The horizontal lines present the average values for each group.

Looking at the session quality performance of the three groups when no adaptation is used, we see strong variations in performance. Starting from left to right: the weak, medium and strong peers, had an average of 43, 35, and 23 stalling events respectively. This performance gap is even more visible for total delay, where the maximum delay of 200 seconds indicates that the peers left the system without watching the whole video due to bad performance. What we see here is the natural effect of correlated performance-resources usually evident in media agnostic VoD systems. This usually leads to excluding peers with weak resources from the system, forcing them to leave the system.

However, already with the addition of the IQA, the slow peers can take part in the system and even have good performance. The slow, medium and fast peers had only 9, 14 and 15 stalling events respectively. The total delay mounted to 25, 37, and 43 seconds for the three groups respectively.

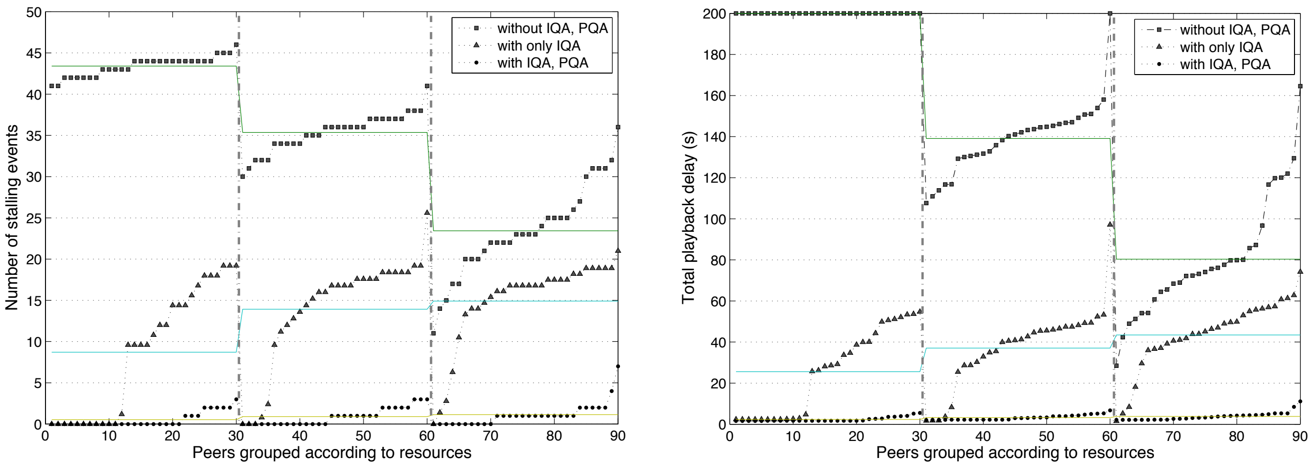
Another interesting improvement gained when using the IQA, is the homogeneous performance for the three groups. We see that having less resources does not affect the session quality, but rather only reduces the video quality. Although the group with lower bandwidth can only receive low quality video, it can nevertheless enjoy continuous playback.

IQA is essential to adapt the system to static resources, however it is not enough as it cannot predict system dynamics. As can be seem from the lowest curves in Figure 5, the performance when using both IQA and PQA was the best. Each peer, irrespective whether slow, medium or fast, witnessed on average 2 stalling events and had 3 seconds of total delay. The PQA, therefore, helps in achieving better session quality and more homogeneous performance across heterogeneous peers.

6.2.2 *Session Quality versus SVC Video Quality*

From the previous evaluation we see the need to have both initial and progressive quality adaptation. For the PQA, the question arises: how often should it be invoked? i.e. how often should each peer adapt to system variations? To better understand the effects of this parameter and to also understand the trade-offs regarding adaptation dynamics, we evaluate the system for different PQA intervals, namely: 5, 10, 20, 30, 45, 60, and 90 seconds.

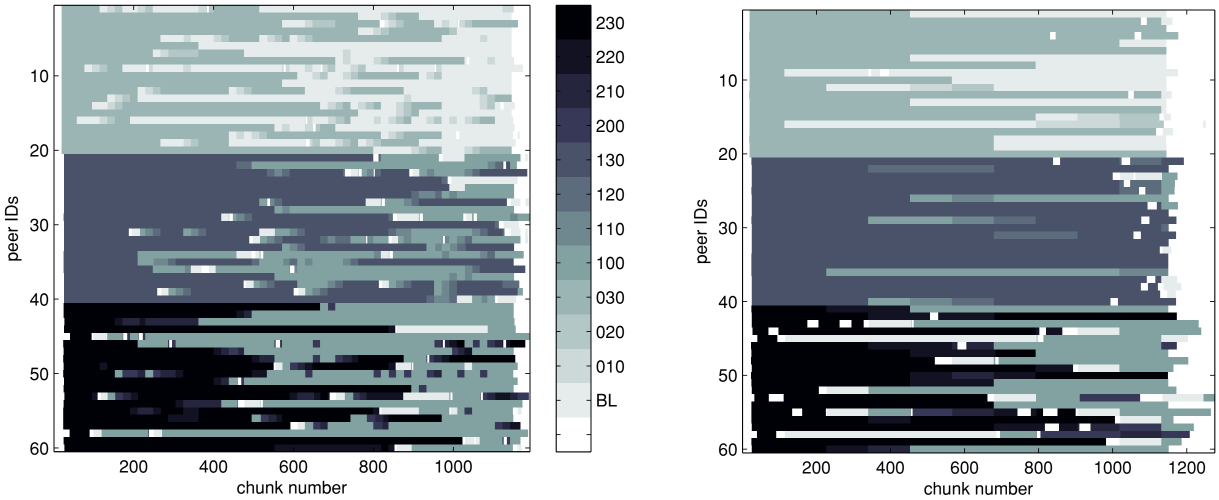
To assess the effect of having different PQA intervals, we



(a) Number of stalling events per peer

(b) Average playback delay per peer

Figure 5: Quality-adaptive streaming using SVC versus streaming using a media agnostic system (no IQA, no PQA). The peers are divided into three groups according to their resources (slow, medium, fast). For each group, the peers are sorted according to the performance metric in the Y-axis



(a) PQA interval T=5 seconds

(b) PQA interval T=30 seconds

Figure 6: Visualization of the layer selection decision for different PQA intervals

present in Figure 6, a visualization of the received layers for the peers when having PQA interval of 5 (Figure 6a) and 30 (Figure 6b) seconds. The peers are grouped according to their resources with strong peers in the bottom (ID range: 40-59), medium peers in the middle (ID range: 20-39), and weak peers in the top (ID range: 1-20). The darker color indicates a higher received layer. For clarity, we present only 20 peers per resource group. We can nicely observe the effect of having different adaptation frequencies on the overall layer decision at the peers.

To better quantify this effect, we now go into more detailed simulation results using the metrics defined in Section 5. Session quality performance (number of stalls, average stall duration, and total relative delay) is presented in Figure 7, while SVC video quality (layer changes, relative received layer) is presented in Figure 8. To further assess the importance of having IQA, we run two sets of simulations, one with IQA and PQA, and one with only PQA. Each scenario is repeated 8 times to exclude any random effects. Standard deviations are shown.

Through the comparison of the session quality for the different PQA intervals, we can see that the more frequently PQA is invoked, the fewer stalls will happen (see Figure 7a). Additionally, the shorter are the stall durations (Figure 7b) and total relative delays (Figure 7c). The reason behind this is that with a larger PQA interval, the peers will be slower to react to system dynamics. Based on this observation, we can conclude that the performance, i.e. the session quality, decreases with an increase of PQA invocation interval. Furthermore, when the IQA is not used, the adaptation behavior is no longer predictable. In the range of 5-45 seconds, the total relative delay is almost the same and then strongly increases for higher values. Therefore, the IQA is necessary, since it already prepares the peer from the beginning for better adaptation to system dynamics.

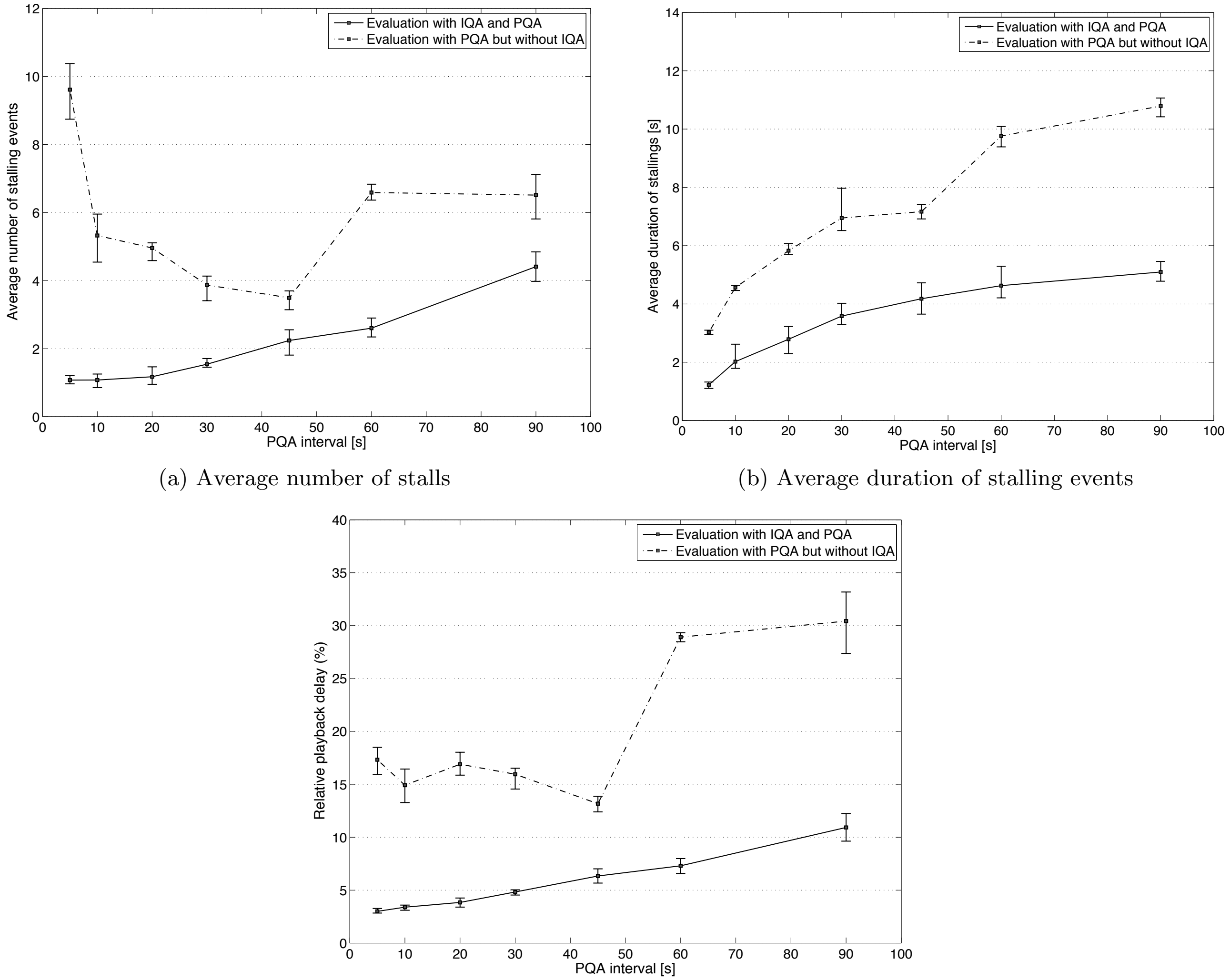
We have also investigated the received SVC video quality. What we desire is a high relative layer level that changes as less as possible, since too frequent layer changes tend to frustrate users [27]. From Figure 8(a) we can see that as the PQA interval grows, the number of layer changes becomes fewer. When the interval is infinitely large, the number of layer changes becomes zero, since the layer selected by the IQA will be used throughout the streaming session. On the other hand, Figure 8(b) shows that the average relative received layer increases with a larger PQA interval. The reason for this is that the PQA usually tends to decrease the layer level to avoid potential stalls during the playback (which explains the better session quality). The more frequently the PQA is invoked, the more layer drops it may cause. Consequently, the average layer level throughout the playback will also decrease. From the results of Figure 8, we can conclude that the SVC video quality for the peers increases with a larger PQA invocation interval.

In summary, the relation between session quality and SVC video quality exhibit a trade-off for the different PQA intervals. Therefore, one has to carefully address this trade-off to achieve a compromise between the performance metrics when choosing the PQA interval. Depending on which aspect is more important for the users, the adaptation interval can be adjusted accordingly to meet the given requirements or can be even chosen dynamically depending on how dynamic the system is.

7. CONCLUSION

In this paper we have explored the performance, tradeoffs, and impact of adaptation on SVC-based P2P Video-on-Demand. The use of P2P technology to support content delivery is very appealing since it enhances the capacity of the P2P network and thus either increases the achievable bit-rate or allows the system to support more peers. In order to support a plurality of different end-user devices and heterogeneous network resources, we propose to use the H.264/SVC standard with full scalability support.

Our investigations revealed that additional control algorithms are needed to enable an efficient provisioning of resources with good video quality. First, Initial Quality Adaptation (IQA) has to be performed for defining the highest video quality, which can be played back on a device. Further, Progressive Quality Adaptation (PQA) has to be performed

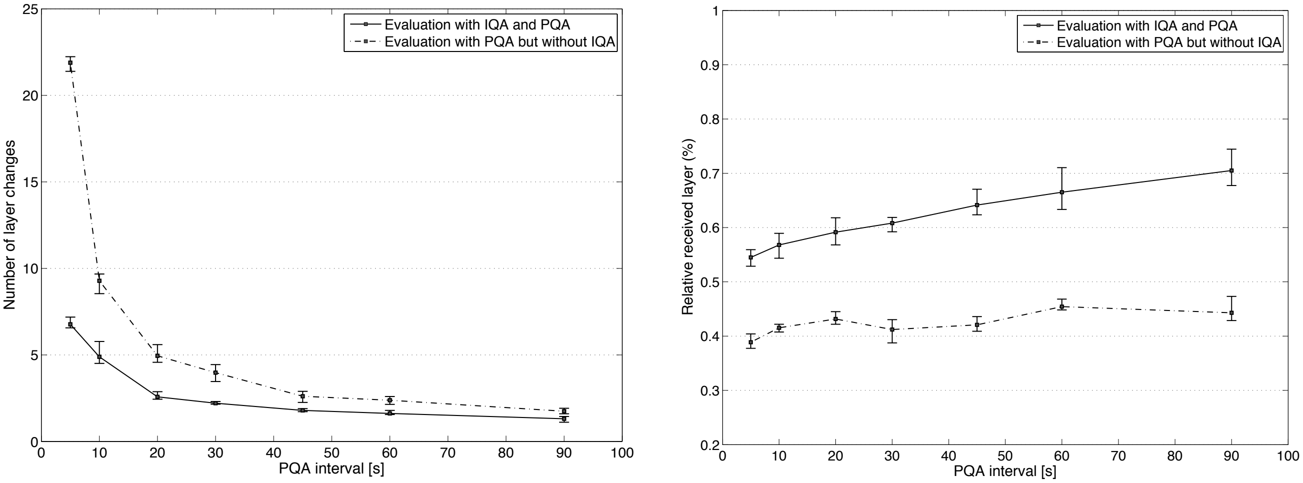


(a) Average number of stalls

(b) Average duration of stalling events

(c) Relative playback delay

Figure 7: Session quality with different PQA intervals



(a) Number of layer changes

(b) Relative received layer

Figure 8: SVC video quality with different PQA intervals

in constant intervals in order to adjust the video quality to the network conditions. An investigation of the length of this interval revealed interesting insight. The longer the PQA interval, the better is the quality of the played out video and the less are quality changes during video playback, but the higher are stalling times, number of stalls and the total delay.

These results are very interesting for providers of VoD services, since they have to take these trade-offs into consideration in order to optimize their system. Further, service providers may offer their customers better quality with longer stalling times if acceptable. These gaps could be filled e.g. with commercial spots in order to increase the revenue of the providers.

Further work will deal with an extended investigation of the system as outlined above. Additionally, we want to map our session and SVC quality metrics to user quality of experience, for example using user surveys. Then, session and SVC video quality indicators would be merged into one, easy-to-optimize quality of experience metric.

8. REFERENCES

[1] O. Abboud, K. Pussep, A. Kovacevic, and R. Steinmetz. Quality Adaptive Peer-to-Peer Streaming Using Scalable Video Coding. In *MMNS 2009: 12th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services*, 2009.

[2] O. Abboud, K. Pussep, M. Müller, A. Kovacevic, and R. Steinmetz. Advanced Prefetching and Upload Strategies for P2P Video-on-Demand. *ACM Workshop on Advanced video streaming techniques for peer-to-peer networks and social networking*, Oct 2010.

[3] O. Abboud, T. Zinner, K. Pussep, S. Oechsner, R. Steinmetz, and P. Tran-Gia. A QoE-Aware P2P Streaming System based on Scalable Video Coding. In *IEEE P2P 2010: 10th IEEE International Conference on Peer-to-Peer Computing*, Aug. 2010.

[4] S. Annapureddy, C. Gkantsidis, P. Rodriguez, and L. Providing Video-on-Demand using Peer-to-Peer Networks. *Proc. Internet Protocol TeleVision (IPTV) workshop*, 2006.

[5] P. Baccichet, T. Schierl, T. Wiegand, and B. Girod. Low-delay Peer-to-peer Streaming Using Scalable Video Coding. In *Packet Video*, 2007.

[6] B. Cheng, L. Stein, H. Jin, and Z. Zhang. A Framework for Lazy Replication in P2P VoD. In *18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '08. ACM, 2008.

[7] H. Chi, Q. Zhang, J. Jia, and X. Shen. Efficient Search and Scheduling in P2P-based Media-on-Demand Streaming Service. *IEEE Journal on Selected Areas in Communications*, 25(1):119–130, Jan. 2007.

[8] Y. Cui and K. Nahrstedt. Layered Peer-to-Peer Streaming. In *13th international workshop on Network and operating systems support for digital audio and video*, NOSSDAV '03. ACM, 2003.

[9] Dissecting the Gap Between Downloading and Streaming Video. Online: http://www.ipsos-ideas.com/article.cfm?id=3804.

[10] M. Hefeeda, B. Bhargava, and D. Yau. A hybrid architecture for cost-effective on-demand media streaming. *Computer Networks*, 44(3):353–382, 2004.

[11] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. A Measurement Study of a Large-scale P2P IPTV System, year = 2007. *IEEE Transactions on Multimedia*.

[12] C. Huang, J. Li, and K. W. Ross. Can Internet Video-on-Demand be Profitable? In *SIGCOMM '07*. ACM, 2007.

[13] R. Iqbal and S. Shirmohammadi. DAg-stream: Distributed Video Adaptation for Overlay Streaming to Heterogeneous Devices. *Peer-to-Peer Networking and Applications*, 2009.

[14] R. Kumar, Y. Liu, and K. Ross. Stochastic Fluid Theory for P2P Streaming Systems. *Infocom*, 2007.

[15] T.-C. Lee, P.-C. Liu, W.-L. Shyu, and C.-Y. Wu. Live Video Streaming Using P2P and SVC. In *MMNS*, 2008.

[16] Y. Liu, Y. Guo, and C. Liang. A Survey on Peer-to-peer Video Streaming Systems. *Peer-to-Peer Networking and Applications*, 1(1):18–28, March 2008.

[17] N. Magharei and R. Rejaie. Adaptive Receiver-driven Streaming from Multiple Senders. *Multimedia Systems*, 11:1–18, 2006.

[18] K. Mokhtarian and M. Hefeeda. Analysis of Peer-assisted Video-on-Demand Systems with Scalable Video Streams. In *MMSys '10: ACM Conference on Multimedia systems*, pages 133–144. ACM, 2010.

[19] M. Mushtaq and T. Ahmed. Smooth Video Delivery for SVC based Media Streaming over P2P Networks. In *IEEE CCNC*, 2008.

[20] A. T. Nguyen, B. Li, and F. Eliassen. Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding. In *IEEE INFOCOM'10*, 2010.

[21] S. Oechsner, T. Zinner, J. Prokopetz, and T. Hoßfeld. Supporting Scalable Video Codecs in a P2P Video-on-Demand Streaming System. In *21th ITC Specialist Seminar on Multimedia Applications - Traffic, Performance, and QoE*, Mar. 2010.

[22] V. N. Padmanabhan, H. J. Wang, and P. A. Chou. Supporting Heterogeneity and Congestion Control in Peer-to-Peer Multicast Streaming. In *IPTPS*, 2004.

[23] K. Pussep, O. Abboud, F. Gerlach, R. Steinmetz, and T. Strufe. Adaptive Server Allocation for Peer-assisted Video-on-Demand. In *24th IEEE International Symposium on Parallel and Distributed Processing IPDPS 2010*, 2010.

[24] J. Reichel, H. Schwarz, and M. Wien. Joint Scalable Video Model JSVM-9. Doc. JVT-V202, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Jan. 2007.

[25] R. Rejaie and A. Ortega. PALS: Peer-to-Peer Adaptive Layered Streaming. In *ACM NOSSDAV '03*.

[26] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9), 2007.

[27] M. Zink, O. Kuenzel, J. Schmitt, and R. Steinmetz. Subjective Impression of Variations in Layer Encoded Videos. In *International Workshop on Quality of Service*, 2003.