QoE-aware Quality Adaptation in Peer-to-Peer Video-on-Demand

Osama Abboud, Julius Rückert, David Hausheer
Technical Report – PS-TR-2012-01

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Electrical Engineering
and Information Technology
Institute of Computer Engineering

Peer-to-Peer Systems Engineering
Prof. Dr. David Hausheer

QoE-aware Quality Adaptation in Peer-to-Peer Video-on-Demand
Osama Abboud, Julius Rückert, David Hausheer
Technical Report – PS-TR-2012-01
http://www.ps.tu-darmstadt.de/

Last revision: February 26, 2012

For the most recent version of this report see
http://www.ps.tu-darmstadt.de/fileadmin/publications/PS-TR-2012-01.pdf

Technische Universität Darmstadt
Department of Electrical Engineering and Information Technology
Institute of Computer Engineering

Peer-to-Peer Systems Engineering
Prof. Dr. David Hausheer

Contents

Abstract

The transmission of video data is a major part of traffic on today's Internet. Since the Internet is a highly dynamic environment, quality adaptation is essential in matching user device resources with the streamed video quality. This can be achieved by applying mechanisms that follow the Scalable Video Coding (SVC) standard, which enables scalability of the video quality in multiple dimensions. In SVC-based streaming, adaptation decisions have long been driven by Quality of Service (QoS) metrics, such as throughput. However, these metrics do not very well match the way human users perceive video quality. Therefore, in this work, the classical SVC-based video streaming approach is expanded to consider Quality of Experience (QoE) for adaptation decisions. The video quality is assessed using existing objective techniques with a high correlation to the human perception of video quality. The approach is evaluated in context of a P2P-based Video-on-Demand (VoD) system and shows that by making peers favor always layers with a high estimated QoE but not necessarily high bandwidth requirements, the performance of the entire system can be enhanced in terms of playback delay and SVC video quality by up to 20%. At the same time, content providers are able to reduce up to 60% of their server costs, compared to the classical QoS-based approach.

1 Introduction

Recent studies show that the streaming of video content has become a dominating part of today's Internet traffic, with a forecast of further increase for the next years [Cis11, San11]. Therefore, the support of an efficient and scalable video streaming is gaining more and more importance.

Besides supporting a large number of users, video streaming systems have to be adaptive to a wide range of heterogeneous devices and constantly changing conditions, such as sudden user fluctuations or network congestion. The provision of appropriate mechanisms for adaptations in these situations is a challenging task. An influence on the provided service quality can be inevitable in some cases. This may become visible to the user in form of degraded video quality or stalling of the video playback. The goal of content providers is to reduce both effects to a minimum in order to maximize the service quality. This requires quality adaptations that allow for a flexible reaction to provide the individual users with a continuous playback and a maximum possible video quality. In this context, the video codec *Scalable Video Coding* (SVC) [SMW07] is especially interesting as it allows for flexible adaptation of the video quality in different dimensions. Furthermore, by limiting adaptation decisions to only compatible quality layers, e.g., in terms of bandwidth or screen resolution, heterogeneous clients with different capabilities can be supported in an efficient way.

For quality adaptation during an SVC-based streaming process, the decisions for appropriate layers are essential. Therefore, *Quality of Service* (QoS) aspects, such as the throughput, are typically taken into consideration to select appropriate quality layers during streaming. However, this approach follows a simplified assumption about video quality and might not directly result in a maximization of the video quality as perceived by a human user. Therefore, the idea proposed in this technical report is to extend SVC-based video streaming with the properties of the human visual system in order to judge the influence on the perceived quality, also referred to as *Quality of Experience* (QoE).

Since the human perception is a complex process and influenced by many factors, only few viable automated metrics, so called objective QoE metrics, exist that can be applied in a general context. At the same time, user studies, which are considered the only reliable alternative to assess perceived quality [Mu09], are not applicable in most technical solutions. Therefore, although automated QoE metrics can only approximate the perception of a user [Win05], they are a promising approach to enable a more user-centric quality adaptation in SVC based video streaming. In the context of the presented approach, the *Video Quality Metric* (VQM) [SMW07], a state-of-the-art objective QoE metric, is applied.

To investigate the impact of using objective QoE metrics for quality adaptation, the focus of this technical report is on a *Peer-to-Peer* (P2P) based *Video-on-Demand* (VoD) scenario. Besides evaluating the adaptation mechanisms on its own, the scenario also allows to study how such adaptations influence the dynamics of a P2P system and the resulting overall performance. For large scale streaming systems, P2P technologies have proven to be a valid alternative to traditional client-server (C/S) solutions. They allow to efficiently leverage resources of end-user devices in order to remove bottlenecks and unburden the content provider from high costs. In principle, the adaptation mechanisms presented in this report could also be applied in a C/S scenario, however, they were designed to meet the specific requirements of a highly distributed scenario, with clients autonomously deciding on adaptations according to their needs.

The remainder of this technical report is structured as follows. In Chapter 3, an overview about existing work in the field is provided, followed by a brief introduction to SVC and VQM in Chapter 2. The QoE-aware adaptation mechanism, which is the main contribution of this work, is introduced in Chapter 4. Subsequently, in Chapter 5, the evaluation of the mechanism is presented. Finally, the paper is concluded in Chapter 6.

2 Background

The video codec *Scalable Video Coding* (SVC) and the *Video Quality Metric* (VQM), which is an objective Quality of Experience (QoE) metric, are two promising technologies that evolved over the last years. SVC and VQM are presented in the following.

2.1 Scalable Video Coding (SVC)

Scalable Video Coding (H.264/SVC or short SVC) is an official extension to the single layer video codec standard *H.264/MPEG-4 Advanced Video Coding* (H.264/AVC). It defines the coding syntax as well as the general schemata and mechanisms for decoding multi-layer videos following the H.264 standard. It was defined as amendment of the H.264/AVC standard document [ITU10]. Schwarz et al. [SMW07, SW08] give a detailed description of the fundamental features and their realization, as defined by the standard. In the following, the most relevant features in the context of this work are briefly summarized.

The most striking property of the SVC standard is its definition of multi-dimensional quality layers. These allow to retrieve so called sub-streams of a video with different visual qualities, from a single SVC encoded source stream. The quality can be varied in terms of resolution, frame rate and signal-to-noise ratio. Within the standard definition, these three possibilities are called spatial, temporal, and quality dimension, respectively. While other video codecs included certain means of scalability, they heavily suffer from a high decoding and encoding complexity as well as a low coding efficiency, in comparison to their single layer variants [SMW07]. Therefore, the goal of SVC was to overcome this problem. This led to very sophisticated mechanisms which are based on the idea to allow for a degradation of quality in any dimension by skipping specific parts of a video stream. The parts which can be skipped are called enhancement layers, while all others belong to the so called base layer and represent the smallest decodable quality level of the stream. The base layer of an SVC stream is always compatible to the H.264/AVC standard, to allow for a minimal quality playback with older video players.

As H.264/SVC is defined as an extension to the H.264/AVC standard, it also inherits all of its encoding techniques, which are applied to allow for scalability as previously described.

The most important property is the concept of three different frame types, which is a known concept since the early versions of the MPEG video codec. These different types of frames are the so called I-, P-, and B-frames [SMW07]. I-frames include the complete information for a picture and can be decoded independently from other frames of the video. P-frames, furthermore, encode information on differences from a temporarily preceding frame, whereas B-frames can additionally encode information on changes compared to consecutive frames. The process of reconstructing the content of a frame's picture from the information encoded by P- or B-frames is called *motion-compensated prediction*. To further reduce the amount of data to be stored for a frame, the information of the frame itself is checked for redundancy within the picture. This is done by using smaller partitions of the picture of a fixed size, so called blocks. This way dependencies within a single picture can be extracted and used to greatly reduce the amount of data to be stored. The process of decoding this information is called *intra-prediction*. It decodes the information by predicting the content of a block through already decoded blocks in the frame's spatial surrounding.

2.2 Video Quality Metric (VQM)

The *Video Quality Metric* (VQM) [PW04] is an objective QoE metric, which was proposed by the *National Telecommunication and Information Administration* (NTIA). It was evaluated by the *Video Quality Expert*

Group (VQEG)[1], a cooperation of experts for the cooperative evaluation of objective *Quality of Experience* (QoE) metrics, during the *VQEG FR-TV Phase II* [VQEG03] and was shown to perform better than any other candidate in this test phase. While this evaluation focused on the quality estimation of videos with typical analog TV resolutions (576i for PAL, 480i for NTSC), Pinson et al., the developers of VQM, state that the metric is applicable to a wide range of other resolutions, codecs, qualities as well as bit rates. In a later work, they showed that it is even applicable for high definition videos [WP07]. This flexibility, as well as its good performance, make the VQM very attractive when deciding between different available objective QoE metrics for a concrete application scenario. According to Engelke et al. [EZ07], the only downside of VQM is that the extracted features from the original content, which are used as reference for the metric computation, require a total of about 14% of the bandwidth of the uncompressed video when transferring them e.g. over a network. This is more overhead than other metrics of the same class produce. For cases where a transmission of the reference information is not necessary, this issue is not a concern. The good performance during the evaluation by the VQEG, led to the standardization of VQM by the *American national Standards Institute* (ANSI T1.801.03-2003) as well as by the ITU (ITU-T J.144, and ITU-R BT.1683). In the context of the presented work, the VQM has been used to realize a QoE-aware quality adaptation mechanism for P2P video streaming.

1 http://www.its.bldrdoc.gov/vqeg/, [Accessed Apr. 10, 2011]

3 Related Work

In the context of this work, there are three different fields of research that have to be considered. They are visualized in Figure 3.1. The most important one is the area of P2P video streaming, followed by approaches that investigate the application of multi-layer video coding for video streaming. The third is the area of video quality assessment and the consideration of *Quality of Experience* (QoE).

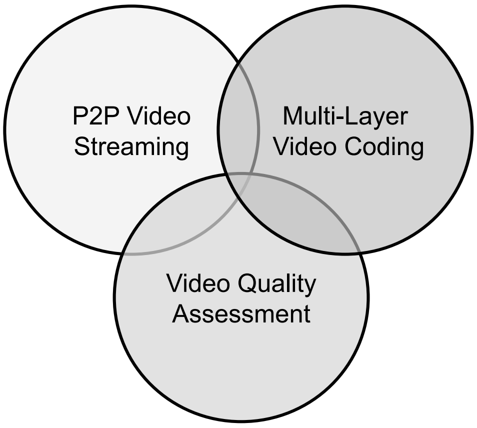


Figure 3.1: The topic of this work intersects three major research fields.

While each of these fields, individually, covers a wide range of ongoing work, their intersection is not considered as often in the literature. The works that are described in the following investigate topics that combine at most two of these fields, never all three. In particular, to the best of the authors' knowledge, there is no dedicated work yet focusing on the investigation of effects that the application of a multi-layer video codec based QoE-aware adaptation mechanism has on the dynamic behavior of a P2P streaming system.

3.1 Focus on P2P Video Streaming and Video Codecs

As previously explained, the streaming of video content imposes special requirements for the data transmission process. Due to the characteristics of nowadays state-of-the-art video codecs, such as MPEG or MPEG2, different parts of a video stream have a different importance for the decoding process. While some data blocks belong to mandatory key video frames, others only transport frames that, in case of a transmission error, may be skipped, while still preserving a decodable video stream. While this characteristic is inherent to single layer video codecs, it is especially true for multi-layer codecs, such as SVC. Due to the realization of different quality dimensions, single transmission blocks could even be assigned more fine granular priorities. Fortuna et al. [FLM+10] used the characteristic of single layer video codecs to realize a pull- and mesh-based P2P streaming system that applies a piece scheduling algorithm and derives priorities from the type of video frames the blocks include.

Other works build upon this work, investigating the special requirements on the scheduling algorithm for multi-layer video content, using SVC [XSGZ09, OZPH10]. In the context of these works, metrics were proposed that allow evaluating the specific characteristics of these codecs. The herein presented work applies some of these metrics, such as the relative received layer ratio. In contrast to these works, the additional use of objective QoE metrics is considered in this technical report to account for QoE characteristics of the streamed content.

In [APKS09, AZPS11], an earlier work of the authors, the impact of quality adaptation mechanisms in pull-, mesh-based streaming systems, using SVC was evaluated. These mechanisms are also considered in the context of the presented system and were further extended to realize more sophisticated adaptation mechanisms.

Besides, there are many further works that investigated the applicability of multi-layer video coding and SVC to P2P video streaming. Cui et al. [CN03], for example, proposed a P2P VoD solution that makes use of layered video codecs to address problems that are introduced by the heterogeneity of peers and the asynchronous video playback process in such a system. Furthermore, Mokhtarian et al. [MH10] derived a model to describe the maximum system capacity of a P2P SVC-based VoD streaming system that provides the peers with a certain minimum video quality. Others proposed concrete systems for both live and VoD streaming, using SVC to address a multitude of different specialized issues, such as low start-up or streaming delays, or a smooth video playback while most of the time trying to assure a certain level of QoS or minimum video quality [BSWG07, MA08]. In contrast to the presented work, these approaches do not consider QoE characteristics of the streamed content.

3.2 Focus on Video Properties and QoE

QoE is influenced by many factors, by the visual quality of the video itself, but also by other properties that are characteristic to video streaming or a result of the quality adaptation mechanisms themselves.

Zinner et al. [ZHAH10] evaluated the impact of temporal and spatial adaptation on perceived quality by using the objective QoE metric VQM to derive dependencies between the quality layers of SVC encoded videos and the perceived quality. Furthermore, they investigated the impact of different upscaling methods, which are used when the spatial dimension of a video is smaller than a device's display size. In the context of the presented work, the derived mappings from SVC layers to objective QoE ratings are used as an input for the later on investigated quality adaptation mechanisms.

Lee et al. [LSE11] executed subjective studies to also derive dependencies between SVC quality levels and the perceived quality. They show that this dependency, on one hand, is influenced by the content type and, on the other, changes with the bit rate condition. While for low bit rates, they identified the spatial layer as the most important factor, for higher bit rates the temporal layer had the biggest impact on the quality ratings by the users. These results could be used as basis for the investigation of a simple adaptation mechanism that prioritizes the dimensions in a certain way during adaptations.

Other works in this direction studied, for example, the impact buffering during the streaming process on the perceived quality [GHP08] or investigated the impact of the layer adaptation process itself on the perceived quality [ZKSS03]. For the latter, characteristics such as the change frequency, the amplitude of changes and the users' preferences for certain paths were evaluated using subjective quality assessment.

3.3 Focus on QoE and (P2P) Video Streaming

The third direction of research is the consideration of QoE in the design of video streaming systems. Most works that share the objective of this work to realize QoE-aware adaptation mechanisms do not specifically aim at a P2P application scenario. Nevertheless, the mechanisms proposed have a high relationship to the presented work and, partially, might also be applicable in a P2P scenario.

Zinner et al. [ZAH$^+$10] propose a framework for QoE management for SVC video streaming systems in general. They investigated the influence on the objectively measured perceived quality from spatial dimension, upscaling method for resolutions smaller than the user device's screen size, degree of network congestion, and the content type. To do so, the pixel-based objective QoE metrics PSNR[Win07] and SSIM[WBSS04] were used. The authors show that SSIM performs better to describe the impact of the adaptations on the perceived quality. Furthermore, they show that it is better to reduce the selected spatial layer in cases of network congestion than to wait for packet losses. Accordingly, the reduction in the video resolution has a lower impact on the perceived quality than distortions induced by lost packets.

Zhai et al. [ZCL+08] propose a centralized live SVC-based video streaming system for wireless, heterogeneous clients, which aims at maximizing the quality perceived by the users. As their scenario imposes special limitations, the authors propose own, low-complexity, objective QoE metrics as well as an adaptation mechanism that uses this metric. The mechanism utilizes the quality estimations of recently received levels to decide on new layers to switch to. To limit the decision range and to simplify the mechanism, the authors propose to first fix the decision for a temporal layer, which, as claimed by the authors, showed to have the biggest impact on the quality. The authors state further, that afterwards the spatial layer should be chosen. The work of Zhai et al., while having the same objective, differs from the presented work in its centralized application scenario with wireless clients. Moreover, the presented work does not aim at estimating the quality of the streamed content on-the-fly at client-side. As a Video-on-Demand scenario is used, there are no real-time constraints and a pre-processing of the content is assumed to do more sophisticated quality estimations.

Furthermore, Kim et al. [KSNR09] propose a centralized SVC-based streaming system that aims at maximizing the QoE of the users. This approach realizes, beside the centralized dissemination, a centralized decision taking approach. Given the currently played layers of a client, a central entity estimates the perceived quality by using an own specialized objective QoE metric, and decides on the potential switch to another layer with a better quality.

Menkovski et al. [MEL10] describe an adaptation mechanism for a distributed streaming system, also covering P2P systems, that aims at maintaining an acceptable video quality for all streaming clients. To distinguish between acceptable and unacceptable quality, the authors directly incorporate user feedback into the system. Using the clients' current QoS parameters and the feedback about the quality, the system builds a continuously updated, distributed decision tree to, firstly, help detecting unacceptable quality for other clients, and secondly, to decide on parameters to improve the quality of these clients to achieve an acceptable level of quality. In contrast to the presented work, this approach does not build on the use of multi-layer video coding but facilitates parameters of traditional, single layer codecs that include limited means of adaptability only. Furthermore, the approach uses no more than two levels of perceived quality which does not account for the characteristics of heterogeneous clients and, if combined with SVC, would not allow to exploit the possible fine granular adaptability of the QoE.

4 QoE-aware Quality Adaptation

This section presents how Quality of Experience (QoE) estimation of SVC encoded videos can be used by quality adaptation algorithms to achieve better system performance. Therefore, QoE-aware quality adaptation mechanisms are proposed, designed, and evaluated, that expand the use of classical QoS driven mechanisms to include other metrics that are closer to how users perceive video quality. What is especially interesting for the performed analysis, is checking whether using estimated QoE in quality layer decisions has any impact on the dynamics of the P2P network and how this influences the performance. For the best of the authors' knowledge, this question has not yet been addressed.

This section is structured as follows. First, Section 4.1 presents a conceptual overview about the quality adaptation approach in general. In a second step, the quality management approach is elaborated on to include QoE ratings in the process of quality adaptation in Section 4.2. Subsequently, the design of the QoE-aware adaptation mechanism is presented in Section 4.3, followed be the two adaptation steps called *Layer Decision* and *Layer Switching* in Section 4.4 and Section 4.5, respectively. For the sake of a clear naming scheme, in Section 4.6, the later on used configuration variants are introduced. Finally, in Section 4.7, this section is concluded with the description on how the objective QoE metric VQM is used for the concrete realization of the quality adaptation mechanism.

4.1 Approach Overview

In order to bring QoE to the quality adaptation algorithms in the SVC-based VoD system, the approach depicted in Figure 4.1 is devised. There are two major steps in the presented system: the *Quality Management* running at the server and the *Layer Adaptation* running at the peers.

The *Quality Management* step running at the server has the main task of calculating the QoE ratings for SVC video sequences to be streamed. This is done by the *Layer QoE Evaluation* module, which calculates the QoE ratings in order to provide them to the peers. As input, it takes the SVC video sequence as well as an *Objective Quality Algorithm*.

The *Objective Quality Algorithm* in the general sense is able to calculate the QoE ratings of a video sequence. Recently, the state-of-the-art in objective quality is able to give very good estimates of the QoE that come close to subjective QoE evaluation methods. Although the actual objective quality algorithm is derived from user studies, such studies are not directly used in system. By executing this algorithm, a QoE rating is calculated for all possible layer combinations. For each SVC file that is offered by the system, respective QoE ratings have to be calculated.

The *Layer Adaptation* module comprises the adaptation algorithms [APKS09, AZPS11]. These algorithms decide on a certain SVC layer given the resources available at the peer. Therefore, these algorithms provide a certain decision on what would be the best choice for blocks to request from other peers given the different resources. The standard adaptation algorithms do not use any information concerning the QoE when deciding on these layers.

The QoE ratings are used by the peers to make certain adaptation strategies. The adaptation strategies, as explained next, address possible opportunities and limitations when deciding on which SVC layer to chose. Since each SVC file has different QoE ratings, the adaptation strategy itself is not static, rather, it will be different for each video file.

One major research question which is addressed in this work is, what is the impact and reaction of the system to a adaptation strategies derived from QoE-ratings?

In the presented system, the QoE ratings are used to select SVC layers that have the maximum quality. Second, the ratings themselves are used in the performance evaluation of peers and the system. Thus, the subtle difference between those two approaches is that the first makes a recommendation on which layer to choose while the second checks how much these recommendations were sustained by the peer. The

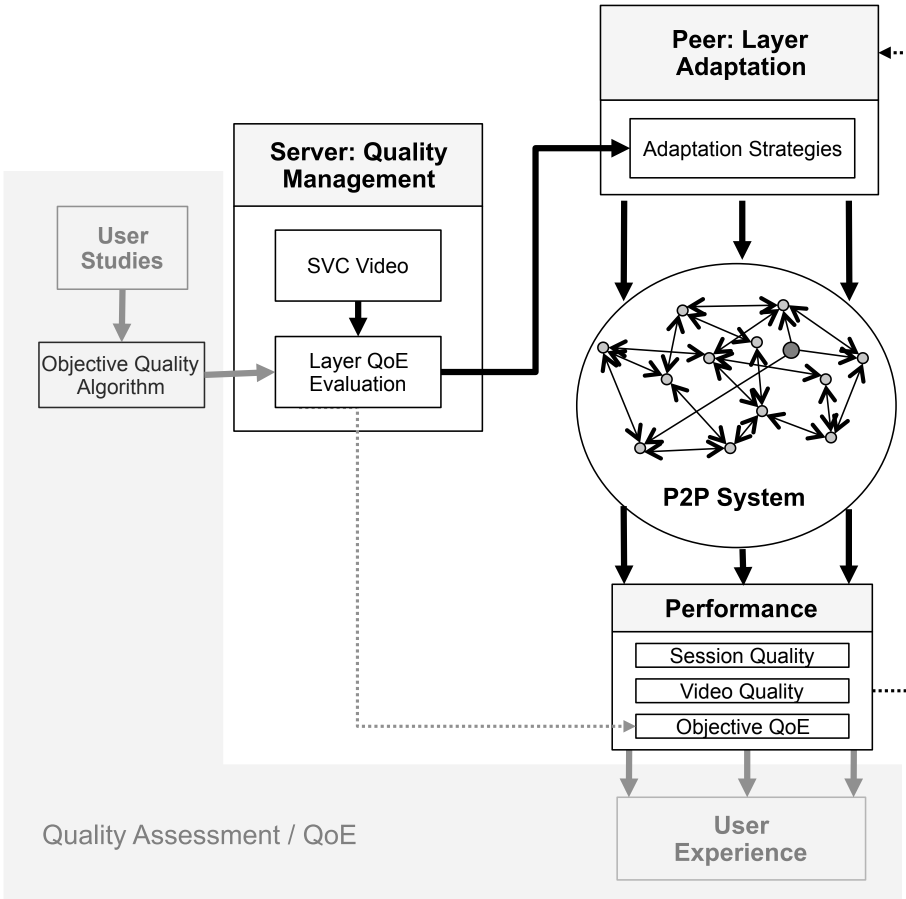


Figure 4.1: Approach for including QoE rating to the quality adaptation algorithms.

main reason why the chosen layers cannot be sustained can mainly be related to the highly dynamical nature of the P2P network where resources are not guaranteed. Nonetheless, as shown later, choosing SVC layers based on QoE metrics helps in making P2P resources more reliable.

The next sections elaborate on the quality management and the layer adaptation modules. The metrics used to evaluate the performance of the system are presented in Chapter 5.

4.2 Quality Management

In order that the peers can take QoE ratings of the SVC video into account, they have to be estimated, derived, and sent to the peers. This is done by the *Quality Management* part of the system running at the server. The details behind this part are given next.

4.2.1 Deriving The QoE Ratings

The QoE ratings are derived using the Objective Quality Algorithm for all possible SVC layer combinations. Although subjective user studies can be performed to estimate the QoE ratings for all SVC layer combinations, this is not feasible as it is not possible to perform user studies for every single offered video content. Therefore, more subtle methods, i.e. objective methods, which can work completely human free, are more viable for such applications. It is worth mentioning that the actual development of objective metrics involves insights gained through studying the human visual system. Therefore, objective QoE methods make it possible to automate the quality management process. This is essential especially for future streaming applications where the amount of video content offered to the users is likely to increase tremendously.

4.2.2 QoE Ratings Model

While in general the system is able to work with any objective quality algorithm for deriving the QoE ratings, some formal requirements on the values of the QoE rating are specified in the following. They should be positive numbers and range between 0 and 1, where 0 represents the best quality and 1 represents the worst quality. Considering two SVC layers having R_1 and R_2 as QoE ratings, if $R_1 < R_2$, then layer 1 has a better quality than layer 2.

4.2.3 QoE Ratings Offline Calculation at the Server

Since all the peers are required to have the QoE ratings and since calculating these ratings is CPU intensive, the server entity has the task to calculate and manage these ratings. Therefore, the content provider needs to calculate the QoE ratings for each video file it is offering to its customers. Since this work focuses on a VoD scenario, this is feasible by preprocessing the video file before its release. The QoE ratings are, therefore, calculated offline, before streaming starts, and is distributed to the peers along with the meta data. This information can be delivered by the tracker or the streaming server. The ratings can also be enhanced by dividing the video file into smaller chunks or scenes and by deriving ratings for each of them. Both approaches are compatible with the presented solution.

4.2.4 Advantages of Objective QoE

This section discusses in detail the choice of objective QoE estimation methods.

Automation of QoE Estimation.
The main advantage of using objective QoE is due to the possibility to automatically process, generate, and manage the QoE of a huge amount of video content. Additionally, it is possible to evaluate the impact of different adaptation algorithms even using simulative evaluation, which is not possible using subjective QoE metrics.

No Seed to Compute QoE at Peers.
Since QoE ratings for all possible SVC layers are computed in advance, the peers do not need to invest their resources in this task. In contrast to the work presented in [ZCL$^+$08], no calculation is required at the peer to derive the QoE ratings. The limitations of the mentioned work is that the peers have to invest much processing power in estimating the received video quality. Evidently, deriving objective QoE ratings, especially reliable ones, is very processing intensive [PW04]. Since a scenario with heterogeneous devices and that respects limited processing capacities is considered, it is essential to minimize the processing footprint of the quality adaptation algorithms. According to [ZCL$^+$08], the peers can only know the quality rating after they have switched to the respective layer. This solution is, therefore, not suitable for assessing the full potential of quality adaptation.

No Need to Send a Reference Video.
The presented mechanisms work by using the sent QoE ratings of the SVC video file without any additional information. The positive side effect of this, is that no additional video reference (whether full or reduced) has to be transmitted to the peers, thereby preserving precious bandwidth for the actual video transmission. The amount of data required for the reference data can be rather large. For example, using VQM along with reduced reference, the amount of overhead can reach up to 14% [EZ07]. Using

such as approach, no reference data has to be submitted. Nonetheless, it is still possible to utilize good estimates about the video quality using the state-of-the-art objective QoE estimation methods.

4.2.5 Multiple QoE Rating Sets

It is worth noting that although the system evaluation is restricted to only one set of QoE ratings for the whole file, the design can still be extended to include multiple sets. This can be beneficial especially for video files that have drastic changes in scenery, for example from low to high motion. In this case, the content provider would calculate separate QoE ratings for each scene in the video sequence.

4.3 QoE-aware Adaptation

This section furthers the quality adaptation mechanisms presented in earlier works [APKS09, AZPS11] by extending the classical QoS-based algorithms with ones that consider the QoE.

Quality adaptation in the VoD streaming system is done in two phases. First using the *Initial Quality Adaptation* (IQA) strategy, an initial quality is selected based on the peer's static resources, for example its bandwidth and screen resolution. After streaming starts, another set of algorithms called the *Progressive Quality Adaptation* (PQA) makes sure that streaming and video playback are continuous. If needed, the PQA would increase or decrease the video quality when the throughput increases or decreases respectively.

To extend the quality adaptation mechanisms with QoE-awareness, the QoE ratings introduced above are used. Both the IQA and the PQA are extended with the intelligence required to use the QoE ratings.

4.3.1 QoE-aware Initial Quality Adaptation

The *QoE-aware Initial Quality Adaptation(IQA)* is, similar to the mechanism presented in [APKS09, AZPS11], executed at the beginning of the streaming session with the distinction of using QoE ratings to decide on the final SVC layer. As presented in Figure 4.2, the input from the layer QoE estimation is used by the final decision module to select the layer.

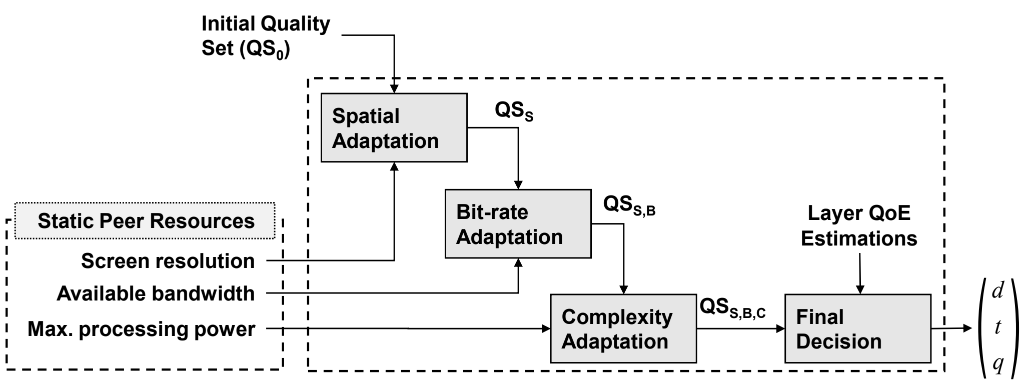


Figure 4.2: QoE-aware *Initial Quality Adaptation* process (Adapted from [AZPS11]).

4.3.2 QoE-aware Progressive Quality Adaptation

The QoE-aware *Progressive Quality Adaptation* (PQA) is again extended from the classical PQA presented in [APKS09, AZPS11], an earlier work of the authors. The PQA has the main task of keeping track of active resources, such as throughput, and react to the changes by increasing or decreasing the video quality. As depicted in Figure 4.3, the QoE-aware PQA has an additional module called *QoE Adaptation*. This module uses the QoE ratings calculated by the *Layer QoE Evaluation* module in order to take the QoE into account while making the final decision. In the basic design, this module would select the layer with the highest QoE rating in order to stream those layers that have the best impact on QoE.

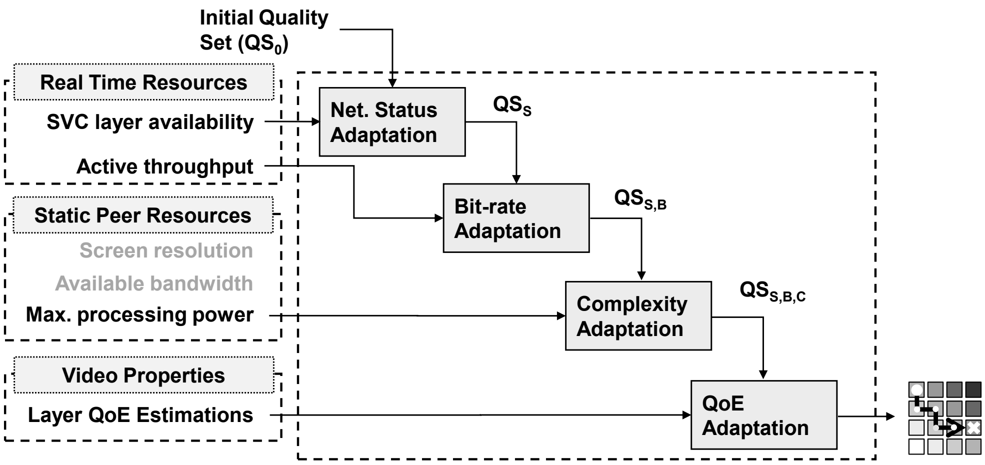


Figure 4.3: QoE-aware *Progressive Quality Adaptation* process (Adapted from [AZPS11]).

Since the PQA is executed periodically, there is time between two executions for the adaptation process. Motivated by the fact that too frequent layer variation can have an adverse effect on the quality of experience [ZKSS03], mechanisms to switch the layer smoothly are presented in the following. The actual layer adaptation is a two step process. The first step is the *Layer Decision* while the second is the *Layer Switching*. The overall logic and design for the two step adaptation as well as their input parameters and outputs are visualized in Figure 4.4.

4.3.3 QoE Adaptation.

QoE Adaptation is performed over two steps, the layer decision and layer switching. Both take the QoE ratings as input in order to select layers with the highest ratings. Those ratings, as explained above, are delivered from the layer QoE evaluation module by the server.

The QoE adaptation is performed as follows. The layer decision is executed on the PQA list to select a new layer termed *target layer*. The layer switching step follows by defining a switching or adaptation path that starts from the current layer and smoothly changes the quality to the target layer.

Layer switching, as presented later in Section 4.5, offers different possibilities to switch from SVC layer *A* to layer *B*. A simple possibility, is to directly switch by making a jump. Another possibility is to allow for smoother switching in terms of keeping the variation of QoE ratings as low as possible.

In the following two subsections, the *Layer Decision* and the *Layer Switching* modules are presented in detail.

4.4 Layer Decision

The *Layer Decision* module has the main task of deciding on a layer that fulfills a certain criteria. The main criteria is to maximize the QoE ratings of the played out video stream. Therefore, this module, given several SVC layer options, chooses the SVC layers that have the highest QoE rating. For the sake of comparability, the three other alternatives are further evaluated. These are the *Simple Layer Decision*, the

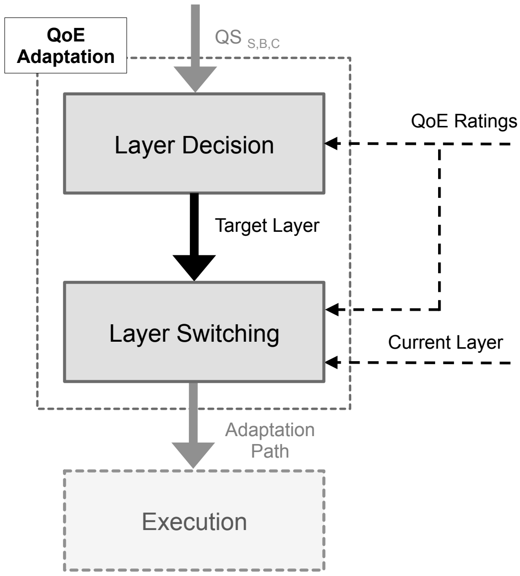


Figure 4.4: Steps of the QoE-aware adaptation mechanism: *QoE Estimation*, *Layer Decision*, and *Layer Switching*.

Maximum Bandwidth Utilization, and the *Prioritized Dimensions*. These different strategies are described in more detail in the following.

4.4.1 Simple Layer Decision (D_{Sim})

The Simple Layer Decision strategy is the simplest layer decision strategy, which was inspired by the related work presented by Oechsner et al. in [OZPH10]. In that work, the authors propose that layer selection follows a slow ramp up strategy, in which the base layer is selected for 10 seconds, then more layers are added along the way. Since the original approach does not involve any quality adaptation, i.e. the selected layers are static and defined before the streaming starts, it is slightly modified to allow for more fair comparison.

First, all peers start by requesting the base layer to allow for a fast start-up time. After that, the peers increase the layer with every progressive quality adaptation step.

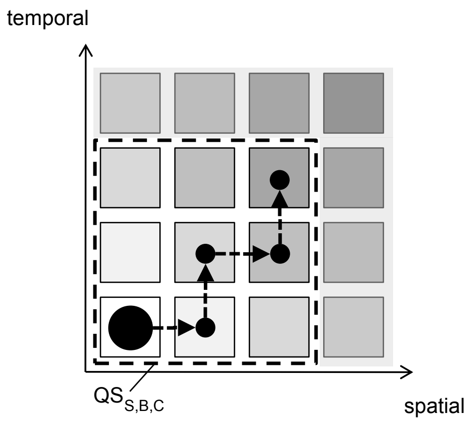
Layers are switched according to the following algorithm. When increasing the SVC layer, the different scalability dimensions are increased in a round-robin manner in the order *spatial*, *temporal*, and *quality*. Always the layer with the lowest value is switched up. If one dimension reaches the limit set by the IQA, it is no longer increased, but the other scalability dimensions are. For switching down the layer, the reversed order for the layers is used, i.e. *quality*, *temporal*, and *spatial*. In all cases, the bit rate of the selected layers cannot exceed a peer's bandwidth limits, otherwise a video stalling may occur.

Figure 4.5 illustrates an example for the simple layer decision strategy. For simplicity of presentation, an SVC file of two dimensions is assumed. The figure on the left shows the adaptation decision when increasing the layer, while the one on the right shows the adaptation decision when reducing the layer.

4.4.2 Prioritized Dimensions (D_{Prio})

The prioritized dimensions strategy works by defining a certain pattern for the layer switching. This is another simple strategy that uses the QoE ratings of scalability dimensions rather than of individual layers.

The prioritized dimensions strategy requires the definition of the order or importance of the scalability dimensions. For example, [ZHAH10] shows that increasing the layer in one SVC dimension has larger



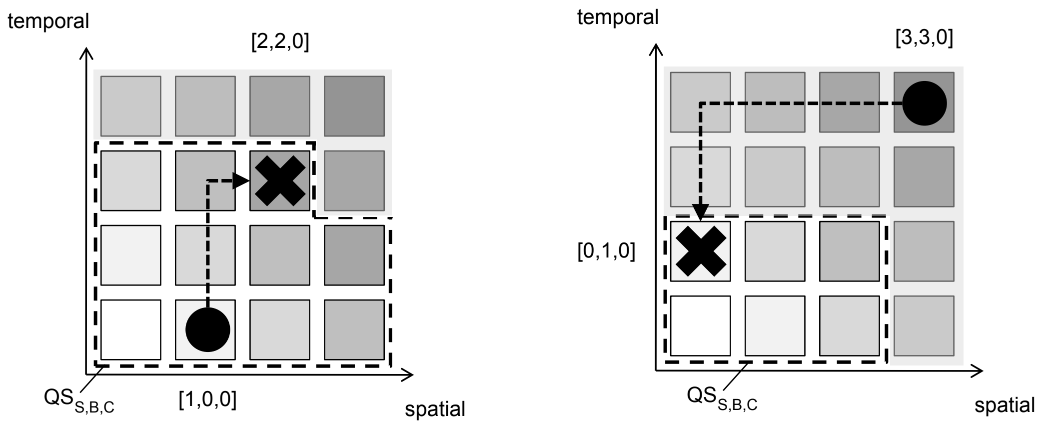
(a) Layer increase

(b) Layer reduction

Figure 4.5: Examples for adaptation decisions using D_{Sim}

impact on the QoE in comparison to other dimensions. The study concludes that it would be best for the QoE to aim at maximizing first the *temporal* dimension, then the *spatial* dimension, and last the *quality* dimension. This strategy adapts one dimension at a time.

The concrete adaptation works as follows. Given the layer from the previous quality adaptation, and as long as resources are enough, the quality is increased in the dimension that has the highest impact on QoE. This is done until a certain dimension is saturated, at which the second most important dimension is increased and so on. It is worth noting that the algorithm still obeys the IQA in terms of selecting only those layers that the peer can actually sustain, whether in terms of bandwidth, screen resolution, or processing capacity, as defined in the $QS_{S,B,C}$ list. Upon scarce resources, the quality is decreased following exactly the opposite order, i.e., first the quality, then the spatial, and last the temporal dimension.



(a) Layer increase

(b) Layer reduction

Figure 4.6: Examples for D_{Prio} with dimension priorities *temporal* > *spatial* > *quality*

Figure 4.6 depicts an SVC video example with four temporal, four spatial, and one quality layers. The figure on the left hand side depicts the layer decision in case of a layer increase from [1,0,0] to [2,2,0]. The dashed polygon indicates the newly supported layers as calculated by the PQA. The actual algorithm is demonstrated using the arrow that is directed from the layer marked with a circle (old layer) to the layer marked with a cross (new layer). Following the general recommendation to first maximize the temporal dimension, this arrow first traverses the temporal dimension until hitting the bounds set by the PQA and then starts traversing the spatial dimension until it reaches its target layer.

Figure 4.6b on the right hand side shows the process of reducing the layer [3,3,0] to [0,1,0]. In this case, first the spatial, then the temporal level are decreased while searching for the layer that fits the new resources as indicated by the dashed polygon.

Although this strategy does not use QoE ratings of individual SVC layer, it still does use the ratings of the dimensions, making this a rather simple but powerful strategy. The newly selected layer will be used as a starting point for later adaptation.

4.4.3 Maximizing the Bandwidth Utilization (D_{Bw})

The goal of this *Layer Decision* strategy is to maximize the bandwidth utilization at the peers. Thereby the peers choose the layer, out of those selected by the PQA (i.e. the $QS_{S,B,C}$ set), that has the highest bit rate.

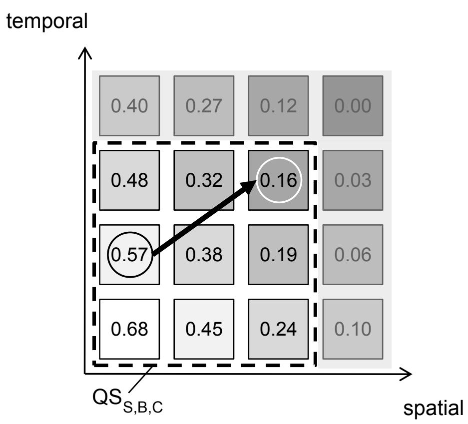
This strategy is not QoE-aware as it does not consider any implication on QoE, rather it focuses on fetching the layers with the highest bit rate. This strategy is used for the sake of comparison and analysis.

4.4.4 Maximizing QoE Ratings (D_{QoE})

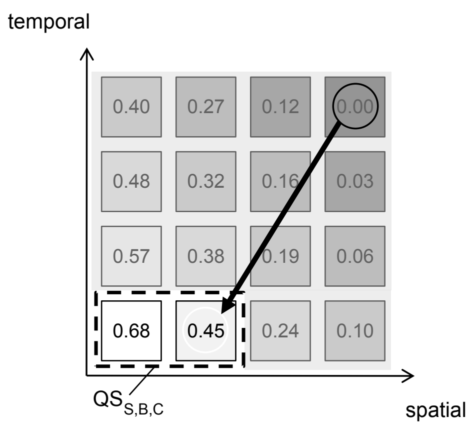
The novel layer decision strategy proposed in this work differs from the other strategies in that is uses full knowledge about the QoE ratings of the different SVC layers during quality adaptation.

Using the information provided by the *Layer QoE Evaluation* module, a QoE rating for each SVC layer is derived. Recall that a QoE rating ranges between 0 (best quality) and 1 (worst quality). Figure 4.7 shows an example based on the Video Quality Metric (VQM). The ratings start off with the lowest value for the highest quality at the upper right most block. The values decrease in any direction towards the base layer, which then has the lowest rating.

Recall that the D_{QoE} strategy selects the layer that maximizes this rating. The algorithm iterates over all layers with the dashed polygon, or PQA chosen layers, and selects the layer that has the highest QoE rating and, therefore, the best video quality.



(a) Layer increase



(b) Layer reduction

Figure 4.7: Examples for D_{QoE}. The numbers present the QoE ratings based on the Video Quality Metric (VQM).

4.5 Layer Switching

Since the time between two adaptation processes can be configured to span several seconds or even minutes, switching from one layer to the other can be done more smoothly by stretching the process over a longer time.

The *Layer Switching* module defines how to switch to a new layer. In other words, given an old layer A and a new layer A as calculated by the layer decision module, the layer switching algorithms define the set of layers that have to be passed when switching from A to B. A smoother switch is motivated by the fact that the perceived video quality can be negatively influenced by too many quality switches [ZKSS03]. Therefore, a stepwise adaptation enables to have quality adaptation with smaller steps in between. To avoid having too much layer variation, the mechanism additionally samples the adaptation path in order to put a limit on the amount of steps until the target layer is reached. This process is described in more detail at the end of this section.

In the following, the concrete realization of layer switching is presented. The mechanism is referred to as *Minimized Absolute Variation in QoE Ratings* (S_{QoE}) and will be described in detail. For the sake of clarity, the configurations without switching is referred to as *Simple Layer Switching* (S_{Sim}). In this case, switching is done in a single step by directly jumping to the adaptation's target layer.

4.5.1 Minimized Absolute Variation in QoE Ratings (S_{QoE})

The S_{QoE} switching uses the variation of QoE rating of the traversed layers as its minimization metric. The goal is to find a path from the old to the new layer which goes through the layers that have the closest QoE ratings. By doing so, the effective quality of the video is changing smoothly between the original and the target layers.

In order to implement the switching strategy, the SVC cube is generalized as a graph. By applying classical, extensively researched algorithms from graph theory, an efficient solution to the problem of minimizing the absolute QoE variation can be achieved.

The above described problem is related to the so called *Single-Source Shortest Paths* problems. Prominent solutions to such problems are the well known Bellman-Ford and Dijkstra algorithms [CLRS01]. The Dijkstra algorithm is chosen in this work as it exhibits lower complexity and can be easily applied to the scenario where the edges between the SVC layers (the variation of bandwidth or QoE ratings) are positive numbers. As the Dijkstra algorithm has been extensively described in literature (see e.g. [CLRS01]), it is not detailed here, interested readers are refereed to the literature.

In the following the old layer is used as source node while the target layer is used as drain or destination. The details on how the graph is derived from he SVC cube is now explained.

4.5.2 Deriving the Graph from the SVC Cube.

Before the Dijkstra algorithm required for the S_{QoE} algorithm can be applied, the graph from the SVC cube is derived.

To start off, each SVC layer in the SVC cube is modeled as an individual node in a graph. Since adaptation is only expected to happen within the PQA-selected layer set, i.e. with the $QS_{S,B,C}$, only the nodes that are within this set are connected. This ensures that only supported layers are reached.

Edges are generated by connecting nodes. Since switching of one layer at a time is considered, each node is only connected to its neighboring nodes that can be reached by adding or removing a layer in any dimension.

Figure 4.8 depicts an example of deriving the graph from a 2-dimensional SVC video with the PQA-selected layer within the dashed polygon.

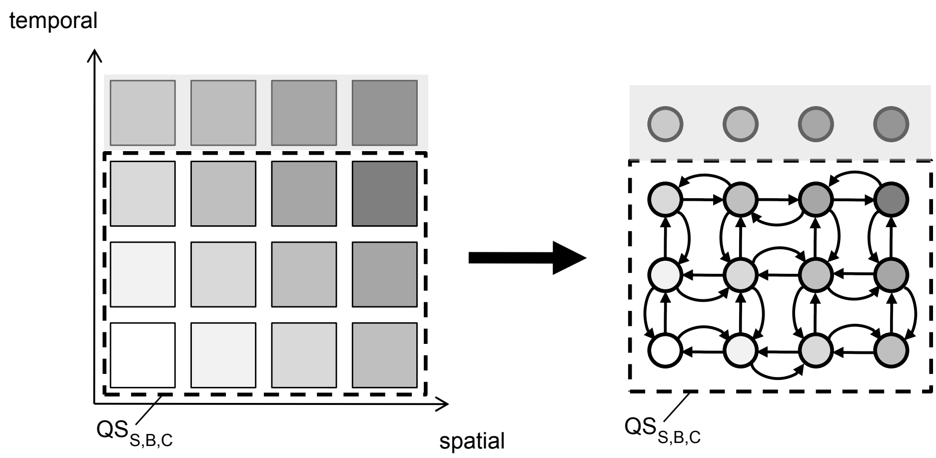


Figure 4.8: The process of deriving the graph from the SVC cube. SVC layers are modeled as nodes and are connected to their direct neighbors in the set $QS_{S,B,C}$.

4.5.3 Deriving the Edge Weights.

The actual weights of the edges that connect the different nodes need to be derived. This is done by using the absolute value of the difference between the QoE ratings of the SVC layers presented by the adjacent nodes. In Figure 4.9 an example on how the edge weights are derived is presented.

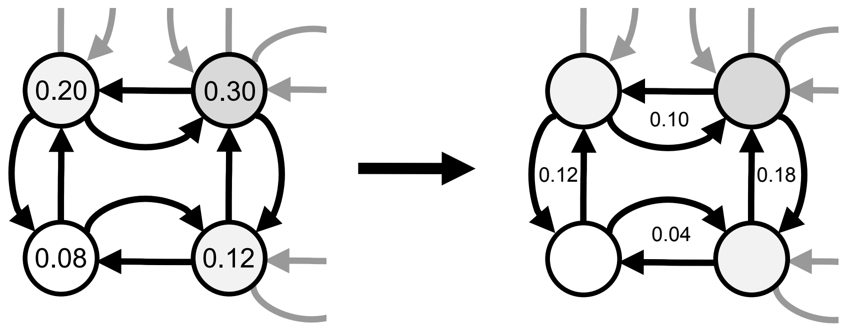


Figure 4.9: The process of defining the edges' weights. The node labels show example quality values for the described SVC layers. An edge weight is the absolute difference between the two adjacent nodes' values.

4.5.4 Minimizing the Variation

The actual calculation of the switching path is performed by applying the Dijkstra algorithm on the calculated graph. Dijksrtra ensures that the retrieved switching path minimizes the switching variation of the QoE ratings. An example derivation of the switching path is depicted in Figure 4.10.

4.5.5 Sampling the Retrieved Paths.

Due to the fact that the main objective of performing *Layer Switching* is to have a smooth transition between an old and a new selected layer, it is essential not to overwhelm the user with too much layer variations on the way [ZKSS03]. Therefore, it might not be that beneficial to switch through too many layers along the switching path. Thus, the path needs to be smpled to reduce the number of jumps.

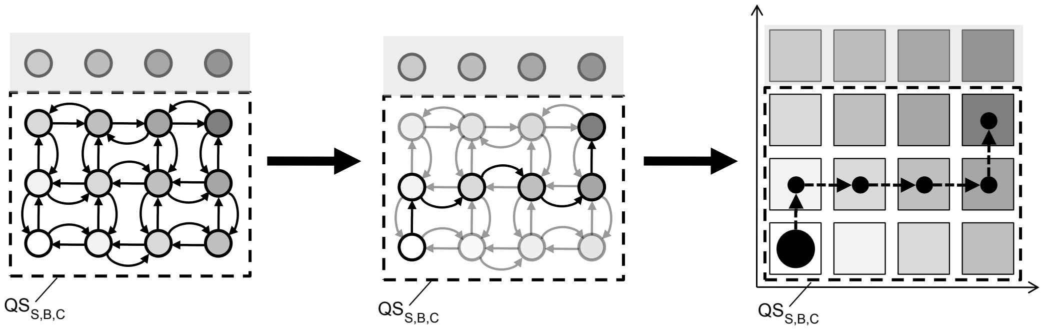


Figure 4.10: The process of retrieving the adaptation path from the graph. The first step is the application of the Dijkstra algorithm, followed by an interpretation of the result as adaptation path.

Since the *Progressive Quality Adaptation* (PQA) mechanism is executed periodically every T seconds (called adaptation interval), the switching from the old to the new layer should be performed before the next PQA operation. Therefore, the length of the adaptation interval poses limitations on the speed at which the switching has to be performed. The larger the adaptation interval, the slower the switching can be performed and vice versa.

Nonetheless, it is important to define a minimum switching time between two layers as to let the user notice a difference when the layer is being switched. Although more specialized subjective tests have to be performed to get a good view on how to chose this value, some studies [NEE+11] suggest to use a few seconds as a minimal value for switching SVC layers.

Therefore, in the simulations a value of three seconds is used as the time between two switches. Since the default value of the adaptation interval is 10 seconds, a switching path involves three adaptation steps by default. Unless stated otherwise, the sampling of the switching path is performed for all strategies.

4.5.6 The Sampling Mechanism.

In the following, the applied sampling mechanism is briefly described. The mechanism defines the intermediate layers to go through, given a previously derived switching path. Therefore, the time each layer has to be sustained is defined as T_{sus}. Since overlapping adaptation operations should be avoided, each operation has to end before the next adaptation is executed.

Given an adaptation path P, as the result of a previous *Layer Switching* process, first the maximum number of adaptation steps N_{max} is derived. N_{max} is calculated as $\lfloor T_{PQA}/T_{sus} \rfloor$. In case $T_{PQA} = 10s$ and $T_{sus} = 3s$, then N_{max} is 3 steps.

If the length of the calculated switching path P is smaller than N_{max}, then no sampling is performed. If the length of the calculated switching path P is larger, then the sampling process chooses N_{max} steps out of the of path P.

In the later case, the step size for the adaptation s_{width} is defined as P/N_{max}. The layer index chosen from the original path P of the i-th entry of the sampled path is determined by: $index(i) = \lfloor s_{width} * i \rfloor$. The last element of the sampled path is always chosen to be the last element of the path P to meet the target layer set by the PQA.

4.6 Configuration of Strategies

In the following, the configuration possibilities are described that are considered later on. Having presented how the *Layer Selection* and *Layer Switching* strategies can use different strategies, an extensive study on the different combinations of the two mechanisms is performed. Therefore, all reasonable combinations of the two strategies are evaluated. Figure 4.11 presents the different possibilities.

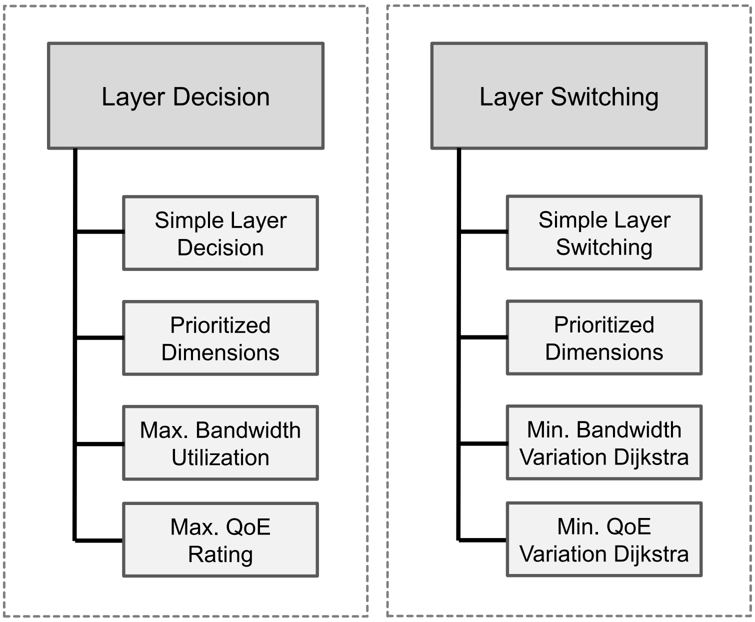


Figure 4.11: This diagram shows the possible configurations for the steps of the adaptation mechanism.

The combinations which are evaluate are shown in Figure 4.1. Next, more details on the different combinations are provided.

Name	Layer decision	Layer switching	QoE-aware IQA
D_{Sim} S_{Sim}	Simple layer decision	Simple layer switching	
D_{Prio} S_{Sim}	Prioritized dimensions	Simple layer switching	
D_{Bw} S_{Sim}	Max. band. utilization	Simple layer switching	
D_{QoE} S_{Sim}	Max. QoE rating	Simple layer switching	X
D_{QoE} S_{Bw}	Max. QoE rating	Min. band. variation Dijkstra	X
D_{QoE} S_{Prio}	Max. QoE rating	Prioritized dimensions	X
D_{QoE} S_{QoE}	Max. QoE rating	Min. QoE variation Dijkstra	X

Table 4.1: Evaluated adaptation configurations.

D_{Sim} S_{Sim} **- Simple Selection.**

This is the reference configuration which employs the simplest selection and switching strategies. As a reference, it allows to better asses the benefits of the other strategies. Many related work [BSWG07, MA08, MH10, CN03, XSGZ09] can be classified as this strategy, as they do not consider any specific QoE ratings during the layer selection algorithms. Since this strategy always starts by selecting the base layer while gradually increasing the layer, it is expected to offer benefits in terms of a faster start-up time.

D_{Prio} S_{Sim} **- Prioritized Selection.**

The *Prioritized Selection* strategy combines the prioritized dimensions layer selection with the simple layer switching strategy. Using this strategy the effectiveness of switching the layer according to predefined priorities of the SVC dimensions is evaluated. As discussed above and presented in [ZHAH10], a default SVC priority order as follows is assumed: spatial, temporal, and quality dimension.

D_{Bw} S_{Sim} **- Bandwidth Selection.**

The *Bandwidth Selection* strategy combines the bandwidth-based layer selection with the simple layer switching strategy. This strategy enables to evaluate how well the system performs in case the peers try to maximize their download utilization. This in general fits to many algorithms in the research field that can be classified as QoS-based. Those pieces of related work are, therefore, compared to the QoE-based adaptation algorithms.

D_{QoE} S_{QoE} **- QoE Dijkstra Switching.**

Finally, the *QoE Dijkstra Switching* strategy combines both the QoE-aware selection and QoE switching mechanisms. This constitutes the main algorithms and techniques that include QoE information into the adaptation algorithms in the presented streaming system.

4.7 Realization using the Video Quality Metric (VQM)

The proposed approach is based on the idea of using objective quality algorithms and metrics to estimate the QoE of SVC video. Thereby, QoE ratings can be calculated offline and without human intervention. Therefore, the state-of-the-art in objective QoE, namely the *Video Quality Metric* (VQM) [PW04] introduced in Chapter 2, is applied.

The ratings generated by the VQM follow the requirements in terms of values, which range between 0 and 1, where 0 indicates the best quality with no visual impairments and 1 indicates worst video quality. This metric is considered the state-of-the-art as it was independently and extensively evaluated and shown to correlate with the human perception of video quality for both TV-like video resolutions [VQEG03] as well as for high definition videos [WP07].

5 System Evaluation

In the following section, the evaluation of the proposed quality adaptation mechanisms is presented. First, the used system capacity model is presented, followed by the general scenario and setup, and the evaluation metrics. Finally, the different evaluated scenarios and results are presented.

5.1 System Capacity Model

To derive scenarios that show the effects of the described adaptation mechanisms, a mathematical model was used to derive the theoretical maximum capacity of a given streaming system configuration. It is based on a model described in [APM$^+$10], an earlier work by the authors. The model defines capacity limits for a streaming system with homogeneous peers and the distribution of non-SVC videos. This model has been extended to account for the heterogeneity of peers as well as the streaming of multi-layer content, such as SVC encoded videos.

It is important to note that for the use of these codecs, in reality, the derived capacity limits are not as static as the ones for single-layer video streaming. The reason is that for a system applying quality adaptation mechanisms, an insufficient system capacity ideally should result in layer adaptations being executed. For such systems, the model describes the theoretical capacity that is necessary to allow all peers to retrieve their maximum layer, as selected during the *Initial Quality Adaptation* (IQA) process. The impact of a reduced system capacity has been evaluated and is discussed in detail in Section 5.4.

For the system capacity model, the following notation is used, which is derived from [APM$^+$10] and has been extended to account for heterogeneous peer configurations and the streaming of layered video content. The concrete notation is as follows:

- S: number of servers
- u_S: server upload capacity
- $G = \{g_1, g_2, g_3\}$: the groups of uploaders, where peers in a group have the same up- and download capacities – The number of groups is fixed to three as this configuration was used for the evaluation. This can easily be changed to account for other configurations.
- U_g, $g \in G$: number of uploaders for each group that have the whole file
- D_g, $g \in G$: number of downloaders for each group having an incomplete file
- u_g, $g \in G$: upload capacity of a single peer in a certain group
- d_g, $g \in G$: download capacity of a single peer in a certain group
- r_g, $g \in G$: initial video bit rate for a group. This is the bit rate of the desired layer which is defined by the result of the *Initial Quality Adaptation* process. It is assumed to be constant within a group and smaller or equal than the download capacity of the single peers in the group ($\forall g \in G : r_g \leq d_g$).
- f: average peer prefetching factor
- g: average peer upload utilization

Furthermore, the following abbreviations were used for a better readability of the equations:

- The total number of up-/downloaders:

$$U = \sum_{g \in G} U_g$$

$$D = \sum_{g \in G} D_g$$

- The average, weighted video bit rate:

$$\bar{r} = \frac{\sum_{g \in G} r_g \cdot D_g}{\sum_{g \in G} D_g}$$

- The average, weighted upload capacity:

$$\bar{u} = \frac{\sum_{g \in G} u_g \cdot (U_g + D_g)}{\sum_{g \in G} (U_g + D_g)}$$

In analogy to [APM+10], this leads to the following dependency for the minimal number of servers:

$$S_{min} = D \cdot \frac{(f \cdot \bar{r} - g \cdot \bar{u}) - \sum_{g \in G} U_g \cdot u_g}{u_S} \tag{5.1}$$

Alternatively, for a given number of servers, this equation can be transformed to calculate of the minimum server capacity:

$$u_S = D \cdot \frac{(f \cdot \bar{r} - g \cdot \bar{u}) - \sum_{g \in G} U_g \cdot u_g}{S_{min}} \tag{5.2}$$

Besides that, it is possible to derive a maximum number of peers, while assuming a given distribution of peers over the groups $g \in G$:

$$D_{max} = \frac{U \cdot \bar{u} + S \cdot u_S}{f \cdot \bar{r} - g \cdot \bar{u}} \tag{5.3}$$

These relations have been used to derive and justify the scenarios described in the following.

5.2 Scenario and Setup

To evaluate the impact of the proposed adaptation mechanisms, a default configuration was defined. It comprises default settings for the host, network, overlay, and video parameters as well as a default pattern for the arrival and departure of peers.

5.2.1 Peer Configurations

The overlay setup comprises a single tracker, nine streaming servers, and 100 streaming clients. The streaming clients are divided into three groups, according to their network and display properties. Table 5.1 gives an overview about the most important peer configuration parameters.

According to the theoretical system model, which was introduced at the beginning of this section, the servers' upload bandwidth capacities are the most important factor for the overall system capacity. To choose a realistic value for this parameter, first the minimum server upload capacity u_S has been determined based on the analytical model. It was then reduced in a stepwise manner to define a scenario that does not overload the server but still generates enough need for quality adaptations. The properties of the modeled SVC video files which are required for the calculation of the minimum server capacity are introduced in the following. Before that, the properties of the modeled SVC video files are introduced. They are required for the calculation.

	Trackers	Servers	$Clients_{slow}$	$Clients_{medium}$	$Clients_{fast}$
Number	1	9	20	50	30
Upload Bandwidth	50 Mbps	25 Mbps	500 Kbps	3 Mbps	8 Mbps
Download Bandwidth	100 Mbps	50 Mbps	2 Mbps	8 Mbps	16 Mbps
Screen Size [Pixels]	-	-	480x270	960x540	1216x684

Table 5.1: The general peer configuration.

5.2.2 Video Properties

The general properties of the three videos, which were used for the evaluation, are listed in Table 5.2. They are retrieved from real SVC video sequences, which were encoded using the SVC reference implementation JSVM[1]. The names used for these sequences are the names of the freely available[2], original test videos. The names are *Blue Sky*, *Crowd Run*, and *Parl Joy*. The properties of the SVC videos are based on the data from previous work by Zinner et al. [ZHAH10]. The SVC videos only include spatial and temporal scalability but no variation in the SNR quality dimension. As the presented approach depends not only on the general properties of the sequences but also on the quality estimations, based on the objective QoE metric VQM, the evaluation was limited to this data. The quality estimations for three videos used are listed in the appendix (Chapter 7).

5.2.3 Minimum Server Capacity

The minimum server capacity can be determined as defined in Section 5.1 above. For the calculation, the extreme case is considered, where all 100 clients are in the leecher state. Leechers are peers that have not yet finished the streaming of the video. Accordingly, the total number of uploaders U is zero and the total number of downloaders D is 100. The number of servers S, as listed in Table 5.1, is nine. The average, weighted upload capacity $\bar{u}$ results in 4 Mbps, given the listed peer capacities. For the calculation of the average, weighted video bit rate $\bar{r}$, the maximum SVC layer for each group of peers has to be chosen. To consider the worst case, it is assumed that for each group one layer is selected that maximizes the bandwidth consumption, while being compatible to the peers' resources. The following calculation is done for the video *Crowd Run*, which is also the default video that was used for the simulations. In this case, the selected maximum layers are the one with layer index 4 for the group $Clients_{slow}$, the one with layer index 13 for the group $Clients_{medium}$, and the one with layer index 19 for the group $Clients_{fast}$ (see Table 5.2 for details). Therefore, the average, weighted video bit rate $\bar{r}$ results in about 950 KBps.

Furthermore, a default average peer prefetching factor f of 1.2 and an average peer upload utilization g of 0.8 is assumed. Using Equation 5.2, the minimum server capacity u_S results in about 61 Mbps. This value defines the minimum required capacity of a single server, assuming all peers retrieve the layer that maximizes their bandwidth consumption, while being compatible to their static properties. For the evaluation of the adaptation mechanisms, a lower default server capacity of 25 MBps is used. The reason for using such a low value is described in the context of the first evaluated scenario. It shows that this configuration leads to a scenario that forces peers to frequently issue quality adaptations because of an overall deficit of resources, but still allows the peers to retrieve the video with a high session and SVC video quality.

[1] http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm [Accessed Aug. 18, 2011]
[2] http://media.xiph.org/video/derf/ [Accessed Aug. 18, 2011]

Layer Index	SVC Level (d,t,q)	Dimension (Pixels)	Frame Rate (fps)	Tot. Bit Rate Blue Sky (KBps)	Tot. Bit Rate Crowd Run (KBps)	Tot. Bit Rate Park Joy (KBps)
0	0,0,0	480x270	1.875	100	83	89
1	0,1,0	480x270	3.75	117	125	134
2	0,2,0	480x270	7.5	134	175	181
3	0,3,0	480x270	15	149	216	197
4	0,4,0	480x270	30	164	235	203
5	1,0,0	640x360	1.875	238	197	196
6	1,1,0	640x360	3.75	279	296	298
7	1,2,0	640x360	7.5	319	423	418
8	1,3,0	640x360	15	356	535	473
9	1,4,0	640x360	30	394	592	488
10	2,0,0	960x540	1.875	345	326	338
11	2,1,0	960x540	3.75	406	486	528
12	2,2,0	960x540	7.5	467	688	779
13	2,3,0	960x540	15	521	881	957
14	2,4,0	960x540	30	575	1,003	1,005
15	3,0,0	1216x684	1.875	484	508	569
16	3,1,0	1216x684	3.75	571	750	891
17	3,2,0	1216x684	7.5	663	1,046	1,328
18	3,3,0	1216x684	15	742	1,334	1,704
19	3,4,0	1216x684	30	820	1,540	1,850

Table 5.2: The properties of the modeled videos as provided in [ZHAH10]. All bit rate values are rounded to the next KBps.

5.2.4 Peer Activities

For the simulative comparison of the adaptation mechanisms in the context of the scenarios described in Section 5.4, an exponentially distributed peer arrival pattern with a mean arrival time of 90 seconds is used. After finishing the download and watching of the stream, peers directly leave the system with a probability of 60%. With a probability of 40%, they stay as seeders for a randomly chosen time interval between 0 and 300 seconds, similar to popular P2P systems [KTP+10]. As the simulated video has a length of 300 seconds, seeders never stay longer than one additional movie length. In Figure 5.1, the progress of an example simulation is shown using these characteristics.

5.2.5 Overlay Parameters

Besides the configuration already described, the streaming overlay is configured using the parameter settings that are listed in Table 5.3. If not otherwise stated in the scenario description, these are the settings applied for the simulations.

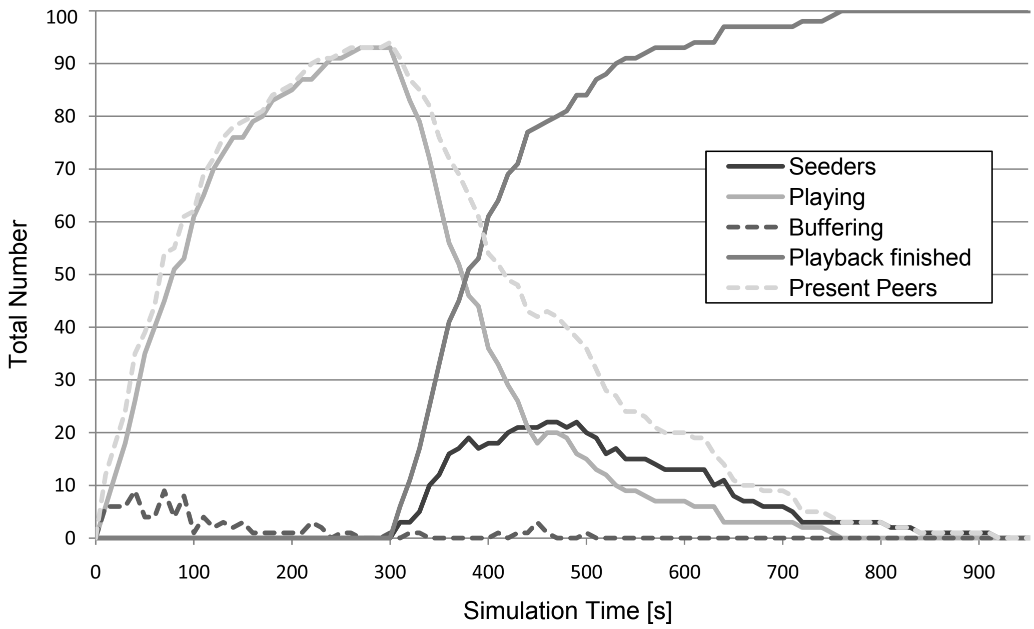


Figure 5.1: An example for the scenario characteristics in terms of peer states and peer arrival pattern.

Parameter	Default Setting
Min. Download Connections	10
Max. Upload Connections	25
Buffer Size	6s
PQA Interval	10s

Table 5.3: The default configuration of the overlay.

5.3 Evaluation Metrics

For the evaluation of the proposed adaptation mechanisms, a subset of metrics that proposed in [AZPS11] and three additional metrics were used. As in [AZPS11] it is herein assumed that these metrics have a direct impact on the perceived video quality. The metrics are described in the following. In analogy to [AZPS11] the metrics described in the following are partitioned into two groups, the *Session Quality* and the *SVC Video Quality* metrics.

5.3.1 Session Quality

This first group includes metrics that describe general properties of the video streaming process. The metrics of this group are independent of the characteristics of the streamed content.

Start-up Delay

The start-up delay is the period of time a peer waits after joining the overlay until enough initial video data is retrieved for the playback to be started. The start-up delay for a peer p is described by Equation 5.4.

$$delay_{init}(p) = t_{playbackStart}(p) - t_{bufferingStart}(p), \tag{5.4}$$

where $t_{bufferingStart}(p)$ defines the time peer p begins with the buffering of the video and $t_{playbackStart}(p)$ the time the playback of this peer starts. It is assumed that a shorter start-up delay directly implies a better perceived quality.

Total Number of Stalling Events

Stalling events happen during playback if video data that is needed for a continuation of the video playback is not retrieved in time. In this case, the video player has to pause the playback and wait for the overlay to provide the missing data. This metric is defined by the number of such stalling events for the overall streaming process, either per peer or in sum for all peers. It is assumed that a lower number of stalling events directly implies a better perceived quality.

Average Stalling Time

This metric describes the average time a peer resided in the buffering state due to an unintended waiting for video data that was not delivered by the overlay in time. It is defined by Equation 5.5.

$$stalling_{avg}(p) = \frac{\sum_{s \in Stalls_p} duration(s)}{|Stalls_p|}, \tag{5.5}$$

where $Stalls_p$ is the set of all stalling events of peer p, and $duration(s)$ is a function that derives the duration of a given stalling event. The initial buffering is not considered for this metric. It is assumed that a shorter average stalling time directly implies a better perceived quality.

Average Total Stalling Time per Peer

The average total stalling time per peer is the total time that a peer resided in the buffering state, including the start-up delay. It is defined by Equation 5.6.

$$stalling_{total}(p) = delay_{init}(p) + \sum_{s \in Stalls_p} duration(s), \tag{5.6}$$

with $delay_{init}(p)$ being the start-up delay, as defined above, $Stalls_p$ the set of all stalling events of peer p, and $duration(s)$ being a function that derives the duration of a given stalling event.

This metric gives an idea of the overall time that peers are waiting for the continuation of the playback. It is assumed that a shorter total stalling time implies a better quality. In [AZPS11] this metric is set in relation to the total video length and call it *relative playback delay*. In the context of the presented work, this metric is used as an absolute measure, while the intention is the same.

5.3.2 SVC Video Quality

The *SVC Video Quality* metrics consider specific properties of the streaming process of layered video content. They are used to describe quality properties depending on the selected layers or the layer switching process itself and are not applicable to single-layer video streaming.

Total Number of Layer Changes

This metric describes the total number of SVC layer changes during the streaming process. It can be used as sum over all peers or as average value per peer. As stated in [AZPS11], a lower number of layer changes directly implies a better perceived quality, according to the results of [ZKSS03].

Change Amplitude

This metric is considered to account for the observation that, besides the frequency of layer changes, the amplitude of a change also has an impact on the perceived quality [ZKSS03]. Therefore, the amplitude of change between two layers is defined as the sum of the absolute difference of these layers' dimensions by Equation 5.7.

$$amplitude(l_1, l_2) = |d_1 - d_2| + |t_1 - t_2| + |q_1 - q_2|, \tag{5.7}$$

where $l_1 = (d_1, t_1, q_1)$ and $l_2 = (d_2, t_2, q_2)$ describe the two SVC layers as triple of their respective spatial, temporal, and quality layer dimensions. According to the results in [ZKSS03], it is assumed that a lower amplitude of a layer change directly implies a better perceived quality.

Relative Received Layer

This metric was introduced in [ZKSS03]. Its objective is to describe a received SVC layer, in relation to the peer's maximum layer. This maximum layer is determined during the *Initial Quality Adaptation* (IQA) process and is the maximum layer that is compatible to the single peer's static resources. The metric is defined by the following equation:

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

with d, t, q being the transmitted and $D_{init}, T_{init}, Q_{init}$ the maximum layer of a peer in the spatial, temporal, and SNR quality dimension respectively.

VQM quality

To gain insight into the perceived quality of the streamed content, the objective Quality of Experience (QoE) metric *VQM* [PW04] was used as introduced in Chapter 2. For the evaluation, it is assumed that quality characteristics of the streamed content are fixed. This means, each SVC layer is assigned a certain objective quality estimation. This way, it is possible to investigate the objective QoE of the SVC layers retrieved by the peers. The concrete quality estimation values of the videos that were modeled for the simulative evaluation were provided by Zinner et al., who used these data in the context of their investigations in [ZHAH10]. The data that are relevant in context of the evaluations are listed in Chapter 7. According to the definition of the VQM quality metric, it is assumed that a smaller value implies a better perceived quality. In Table 5.4, a mapping between VQM ratings and the estimated subjective QoE on the *Mean Opinion Score* (MOS) is given. Pinson et al. [PW04] state that such a mapping is valid and Zinner et al. [ZHAH10] listed this concrete mapping as part of their work.

VQM	MOS
< 0.2	5 (excellent)
≥ 0.2 & < 0.4	4 (good)
≥ 0.4 & < 0.6	3 (fair)
≥ 0.6 & < 0.8	2 (poor)
> 0.8	1 (bad)

Table 5.4: Mapping of VQM rating to subjective QoE (Adapted from [ZHAH10]).

As this data for the QoE estimation are also used in context of the proposed QoE-aware adaptation mechanism, they were only used in the context of the evaluation to verify the mechanisms' effects on this metric and to allow for a comparison to mechanisms that do not use these data as basis for adaptation decisions. To verify that the proposed QoE-aware mechanisms actually lead to an improved QoE would require to conduct subjective user studies.

5.4 Scenarios and Evaluation Results

In the following, the evaluation of the mechanisms presented in Chapter 4 are shown. This comprises four different scenarios that are used to investigate the impact of different configurations and aspect of the adaptation mechanisms.

5.4.1 Scenario 1: System Capacity

The first scenario is used to investigate the impact of a change to the system capacity on the streaming process. The system capacity is dependent on several factors. The most important one is the streaming servers' upload bandwidth capacities, which is herein used as the parameter to change the system capacity.

In subsection 5.2.3, a theoretical minimum for single server capacities has been derived for the presented default system configuration. For this purpose the system model was used that was presented in Section 5.1. Therefore, the worst case was assumed, namely every streaming client aiming at retrieving the SVC layer that maximizes its download bandwidth utilization, while assuring a compatibility of the layer with the client's static resources. By guaranteeing the provision of this minimum server capacity to the system, theoretically, all peers should be able to retrieve their selected maximum SVC layer without any need for progressive quality adaptations. While this might hold for the theoretical model, for a real system this is hard to achieve. The reason is that the theoretical model abstracts the complex connection and download management mechanisms of the streaming overlay. Due to the inherently distributed character of these mechanisms in a P2P streaming system, it is not trivial to allow for a maximum utilization of this theoretical capacity.

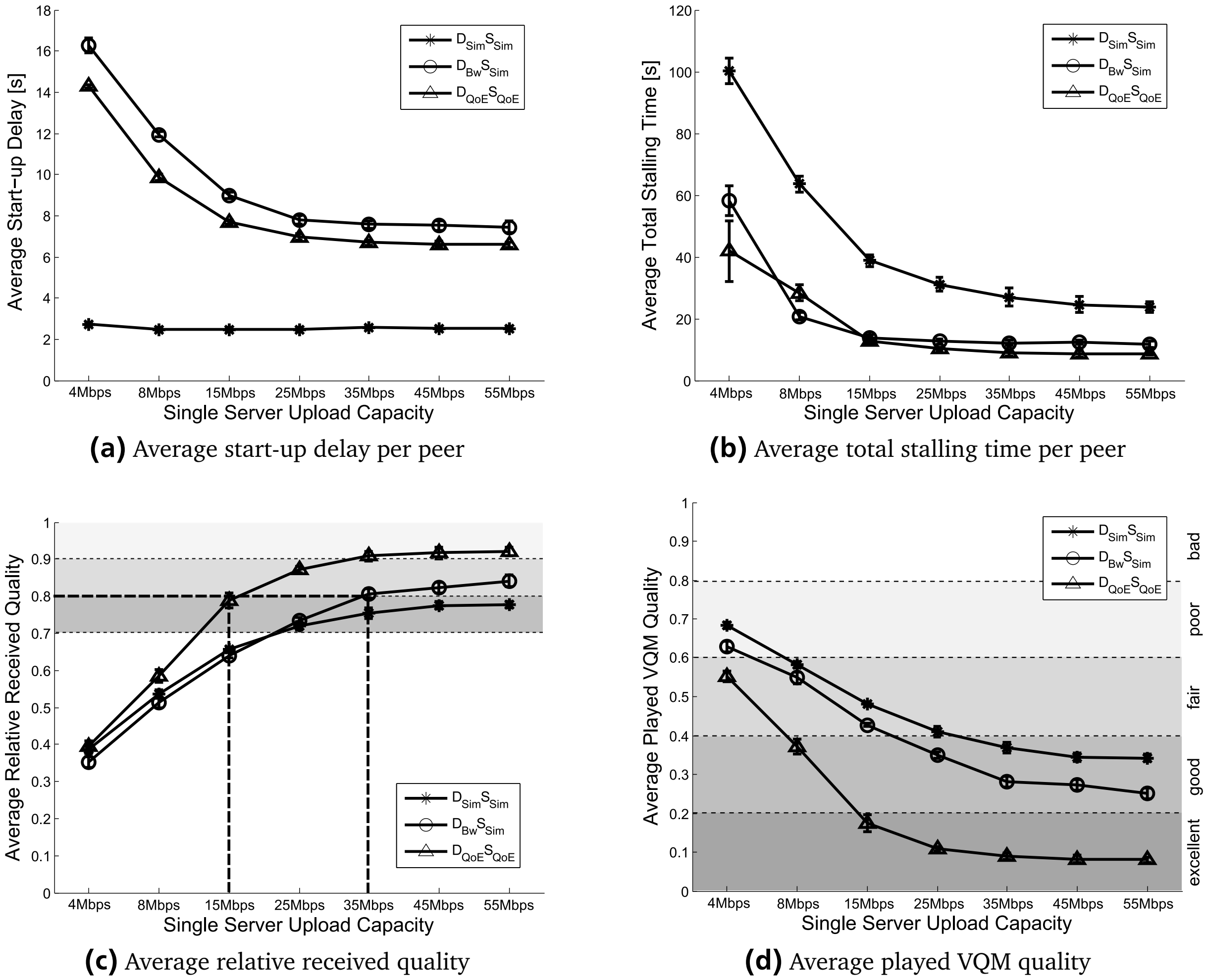
As for the further investigations, the goal is to compare different adaptation mechanisms, the intention is to configure the system not to be able to provide the full capacity to its clients in order to generate a high need for quality adaptations. To find an appropriate configuration that, on one hand, generates a need for adaptations but, on the other hand, does not result in a completely overloaded system, in this first scenario, the server capacity was changed in a stepwise manner to decide on a good default value for this parameter for further simulations. Furthermore, these investigations also aimed at understanding the requirements on the capacity provisioning by a content provider and should help to decide which capacity has to be provided to gain a certain service quality, with a special focus on perceived video quality.

To evaluate the impact of the stepwise change of the streaming system capacities, the upload bandwidths of the streaming servers was used as main parameter. As a starting capacity, single server upload bandwidth capacities of 4 Mbps were used and increased in a stepwise manner up to 55 Mbps, which is just below the derived theoretical value of about 61 Mbps. Independent simulations were conducted for each bandwidth and adaptation mechanism configuration.

For the evaluation, the session metrics covered the start-up delay and the average total stalling time. For the SVC video quality the average relative received quality as well as the average played VQM quality were used. The results for these four metrics are depicted in Figure 5.2.

Analysis

The following can be observed for the session quality: for the higher capacities of 25 Mbps and more, the adaptation variants $D_{Bw}S_{Sim}$ and $D_{QoE}S_{QoE}$ show an almost stable average start-up delay of less than 8 seconds, with $D_{QoE}S_{QoE}$ having a constantly shorter delay by about one second. For lower capacities,



(a) Average start-up delay per peer

(b) Average total stalling time per peer

(c) Average relative received quality

(d) Average played VQM quality

Figure 5.2: Scenario 1: Comparison of the session and SVC video quality for changing system capacities.

i.e. 25 Mbps and below, a steady increase of waiting time can be observable, finally reaching a doubling of the previously observed 8 seconds at the lowest capacity level. $D_{Sim}S_{Sim}$, in contrast, shows a stable start-up delay for all configurations with a value around 2 seconds. This seems to be the main strength of this mechanism and stems from the mechanism's characteristic to always start the streaming process with the retrieval of the content's base layer. This is done at the beginning of the streaming process as well as after experiencing a stalling event. The fact that the base layer has the lowest total bit rate of all SVC layers and because of its constant replication by all streaming peers, this mechanism seems to allow for a highly stable start-up delay even with a low system capacity.

Although the $D_{Sim}S_{Sim}$ mechanism exhibits desirable properties concerning the start-up delay, the simulations show that for the total stalling time, this mechanism performs not as well as the other variants. Starting with a total stalling time of 100 seconds for 4 Mbps, it decreases to a value just above 60 seconds for 8 Mbps, to about 40 seconds for 15 Mbps, and about 30 seconds for the maximum capacity. $D_{Bw}S_{Sim}$ and $D_{QoE}S_{QoE}$, in contrast, both show a more stable stalling time for capacities above 15 Mbps. For the lowest capacity level of 4 Mbps, both show higher total stalling times of about 50 seconds for $D_{Bw}S_{Sim}$ and 40 seconds for $D_{QoE}S_{QoE}$.

For the SVC video quality the following observation can be made: $D_{Sim}S_{Sim}$ and $D_{Bw}S_{Sim}$, although different in detail, perform almost similarly for the relative received quality. For the highest capacity level $D_{Sim}S_{Sim}$ reaches a value of 70%, $D_{Bw}S_{Sim}$ even 80%. Both show slightly lower values for the capacity levels of 45 Mbps and 35 Mbps. For all other capacities, they show almost the same behavior.

For the lowest capacities, a relative received quality of less than 40% can be observed. $D_{QoE}S_{QoE}$, in contrast, shows a stable quality of about 90% for capacities above 25 Mbps. For 15 Mbps the observed quality is at about 80% for 15 Mbps. For lower capacities, it shows an almost similar performance like the other two variants. At the same time, the $D_{QoE}S_{QoE}$ shows a very stable average played VQM quality, with values below 0.2, which maps to an excellent estimated perceived quality on the MOS scale, for all capacities of 15 Mbps and above. For the lower capacity levels, the average played VQM rating is much lower and only maps to the good level of estimated perceived quality on the MOS scale for 8 Mbps and the fair level for 4 Mbps. $D_{Sim}S_{Sim}$ and $D_{Bw}S_{Sim}$ generally exhibit a much lower VQM quality, with the best value being about 0.25, which maps to a good estimated perceived quality on the MOS scale, for $D_{Bw}S_{Sim}$ with a capacity of 55 Mbps. For the lowest capacity level of 4 Mbps both show only a poor estimated perceived quality on the MOS scale.

It can conclude that using the QoE-aware mechanism a great reduction of server capacities is possible in comparison to the non-QoE-aware mechanisms, while providing the same level of relative received quality and VQM ratings. To provide a high average relative received quality of e.g. 80%, a reduction of the server capacities to about 15 Mbps is possible, in comparison to 35 Mbps that would be needed to provide the same quality with the $D_{Bw}S_{Sim}$ variant. This is a relative reduction of almost 60%. For the average played VQM quality, the QoE-aware mechanism was able to provide average ratings that map to an excellent level of quality on the MOS scale. This is even an improvement in comparison to the other mechanisms at 35 Mbps. They showed worse VQM ratings, mapping to the next lower quality level on the MOS scale. At the same time, the QoE-aware mechanism showed a stable session quality with an average start-up delay below 10 seconds and an average total stalling time of less than 20 seconds. For the content providers, these observations show great opportunities for cost reductions. They could define a threshold for the relative received quality, or even better for the estimated perceived quality in terms of the VQM rating, to provide additional server capacities on demand, if a certain critical limit is about to be violated.

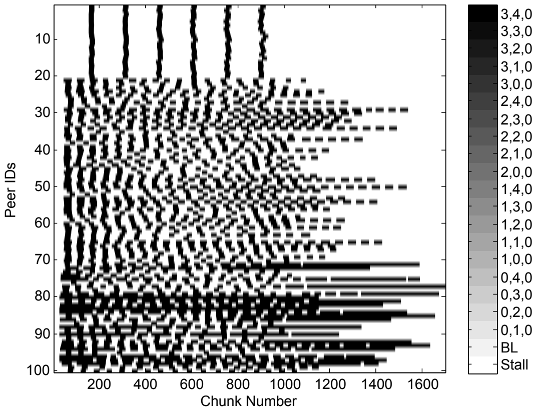
5.4.2 Scenario 2: Adaptation Degree

The second scenario intends to provide an insight to the potentials of quality adaptations in general. Therefore, three different system configurations are used and compared: A first one, without any quality adaptations, a second one that is applying the *Initial Quality Adaptation* (IQA), and a third configuration that uses the full QoE-aware adaptation mechanism $D_{QoE}S_{QoE}$.

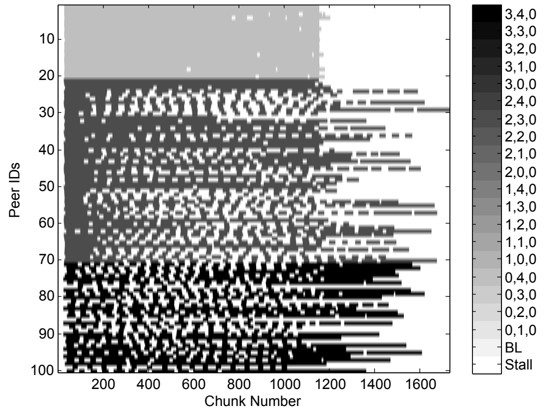
The first does not use any means of adaptation. All peers retrieve the full quality of the video as it would also be the case for single-layer video content. The second degree uses the concept of *Initial Quality Adaptation* (IQA), as described in Chapter 4. It defines a maximum SVC layer that is compatible to the peer's static resources. In this case, without any further progressive adaptation, the peer then would request this maximum layer throughout the whole streaming process. The third degree of adaptation applies all means of adaptation, an IQA process to define a maximum compatible layer and a *Progressive Quality Adaptation* (PQA) process to react on temporal resource bottlenecks. In addition, both the IQA and the PQA mechanism make use of the QoE characteristics of the streamed content to maximize the estimated perceived quality rating.

Besides these different degrees of adaptation, the default configuration parameters are used. In addition, upper limits for the start-up delay and the total experienced stalling time for all peers were defined. It is assumed that a peer immediately leaves the system, due to the low service quality, whenever the start-up delay exceeds two times the total video length or if the additionally experienced stalling time exceeds half of the total video length. While for a good choice for these two ratio values further investigations might be necessary, it proved to be important and reasonable to define upper limits for them. Otherwise, especially for the first configuration, the total stalling time for many peers would have substantially exceeded the total video length.

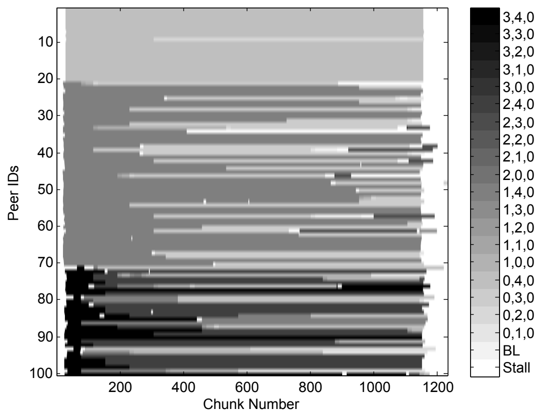
In Figure 5.3, the progress of streaming is visualized. It includes information on each peer's selected SVC layers and its stalling events. Every horizontal line represents the streaming progress of a single peer. According to the three groups of streaming clients, the first 20 lines describe the peers of group $Clients_{slow}$, the next 50 the peers of group $Clients_{medium}$, and the last 30 the peers of group $Clients_{fast}$.



(a) Played layers: without adaptations



(b) Played layers: with initial quality adaptation only



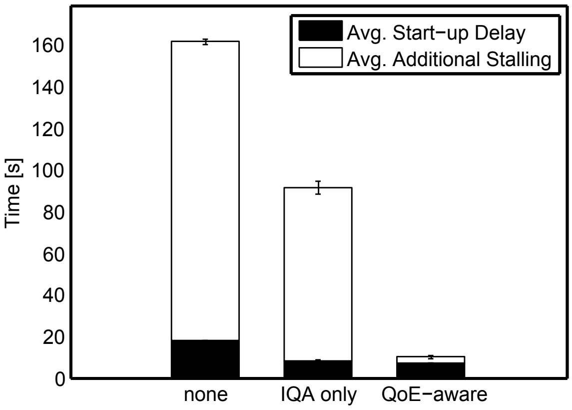
(c) Played layers: with full QoE-aware adaptation, including IQA and PQA

Figure 5.3: Scenario 2: Played SVC layers for the three degrees of adaptation.

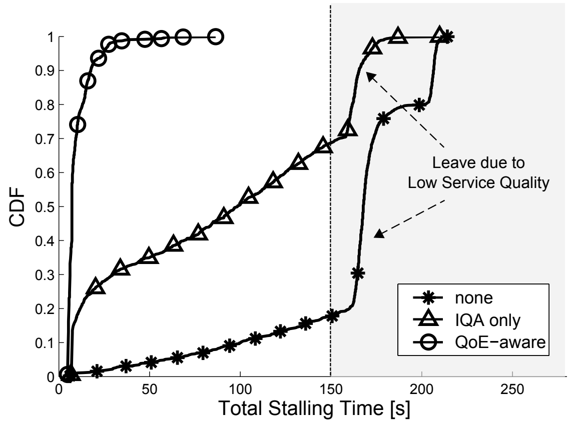
Analysis

For the first variant, a massive domination of white areas becomes apparent in Figure 5.3, indicating a streaming process with a large number of stalling events. For the second variant, the differently selected SVC layers per peer group are visible. Furthermore, the white areas, and with it the number of stalling events, are greatly reduced. Especially for the slower peer's streaming progress, this makes a great difference. For the third variant, the different initial selections of SVC layers are also visible. In addition, changes of the selected layers during the streaming process are observable, indicating adaptations to temporal bottlenecks. This seems to prevent most of the stalling events that are inherent to the other two variants, allowing for a more continuous playback.

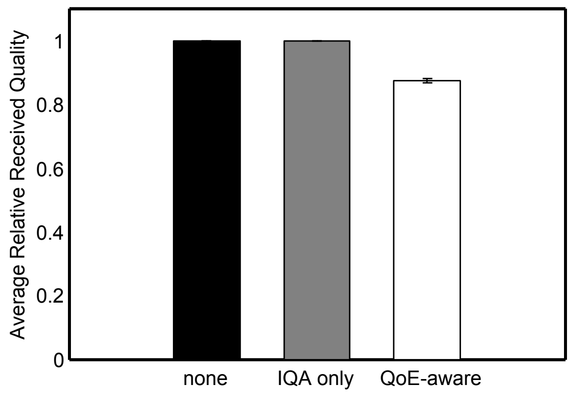
The full extent of differences between the three variants becomes apparent when considering the session and video quality metrics, which were defined earlier in Section 5.3. In Figure 5.4, the most important metrics for this comparison are shown, namely: total stalling time, average received quality, and average played VQM quality. The previously visually observed difference in the stalling events is confirmed by the first graph, Figure 5.4a, comparing the average total stalling times of the peers. For



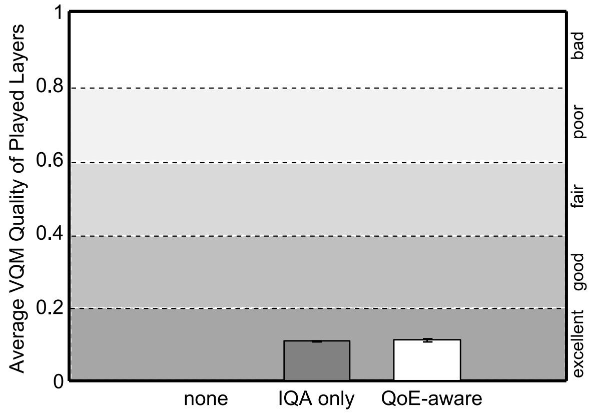
(a) Average total stalling time per peer



(b) Average total stalling time per peer CDF



(c) Average relative received quality



(d) Average played VQM quality

Figure 5.4: Scenario 2: Comparison of session and SVC video quality for different degrees of adaptation.

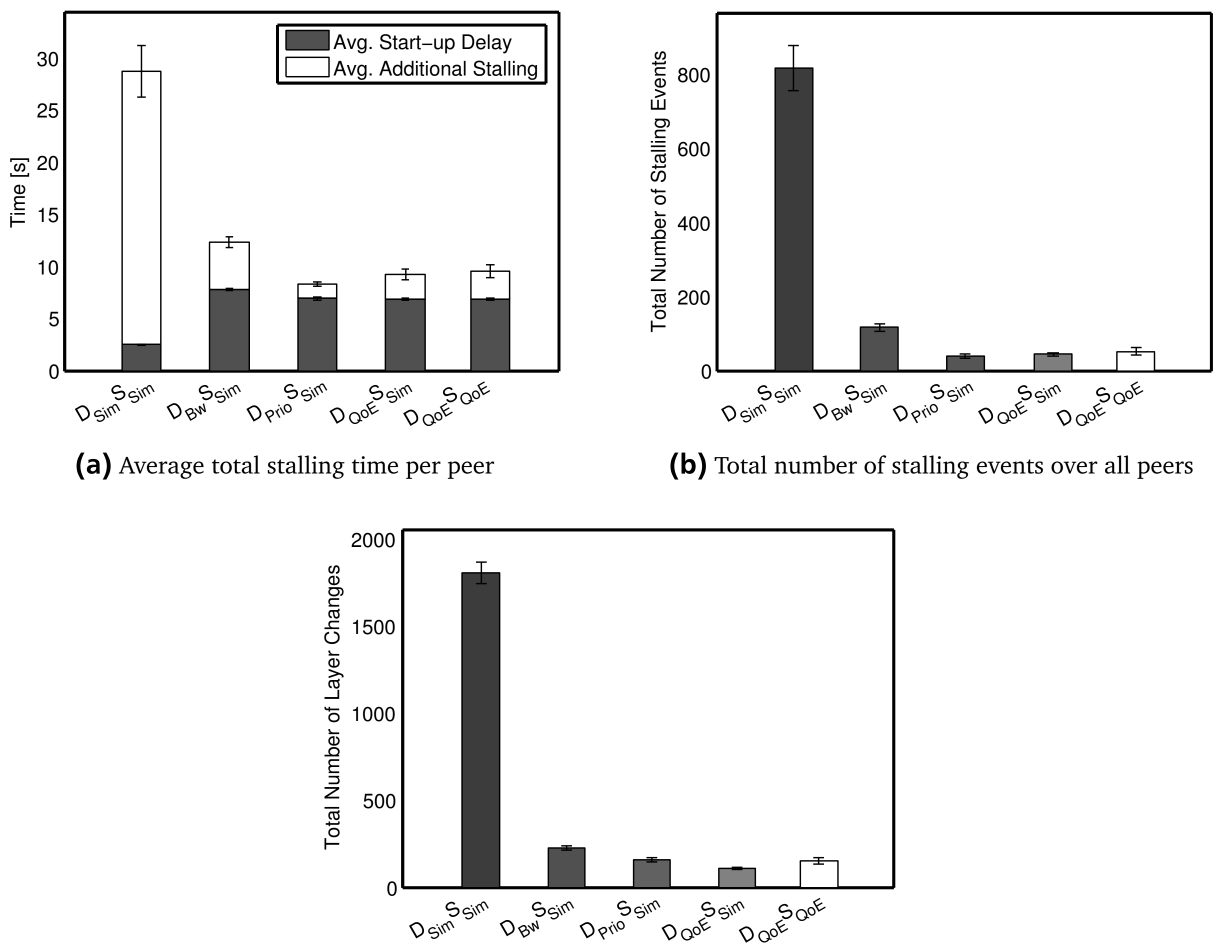
the first variant, peers experienced a total of about 160 seconds stalling, including the start-up delay, whereas for the second variant this delay could be reduced to about 90 seconds and in the third variant to even about 10 seconds. Furthermore, in Figure 5.4b, the *cumulative distribution function* (CDF) of this metric shows that a large fraction of the peers for the first two variants have a total stalling time just above 150 seconds. The reason for this is the defined upper limit for the experienced stalling time, after which a peer leaves the system due to the low service quality. For the simulations, this limit was set to 50% of the total video length, i.e. 150 seconds. This implies that the total stalling time for many peers would have been higher without this upper limit, making the differences even more significant. For the variant without any adaptations, more than 80% of the peers experience a total stalling time above this limit. Using the IQA, this fraction could be reduced to a value of 30%, while with additional progressive adaptations, none of the peers experienced a total stalling time larger than 100 seconds. In this case, none of the peers left the system due to a low service quality.

In Figure 5.4c and Figure 5.4d, the SVC video quality is considered. The average received quality for the first two variants is 1.0, which means that peers always played exactly their maximum SVC layer. Given the intentional lack of progressive adaptations, this value has to be 1.0. In the first case, all peers played the full video quality, whereas in the second case the IQA defined the retrieved layer, which a peer then retrieved with its full quality. For the QoE-aware adaptation mechanism, in contrast, the relative received quality is just below 90%. The VQM quality shows a value of 0.0 for the first variant, indicating the maximum estimated perceived quality, while for the second and the third variants a reduced quality of about 0.1 on the VQM scale was mesaure which maps to an excellent estimated perceived quality rating on the MOS scale, was measured. Although the relative received quality of the QoE-aware variant was lower than for the second variant, its layer selection mechanism allowed to retrieve the same estimated perceived quality, while greatly improving the session quality.

In sum, the constant level of estimated perceived quality and at the same time the substantial improvement of session quality and number of leaving peers, as achieved in the simulations, shows the great potential of progressive quality adaptations.

5.4.3 Scenario 3: Adaptation Variation

The third scenario is intended to provide a comparison of the proposed adaptation mechanism configurations. Therefore, five different configurations are simulated using the default parameter configurations. The used variants are: $D_{Sim}S_{Sim}$, $D_{Prio}S_{Sim}$, $D_{Bw}S_{Sim}$, $D_{QoE}S_{Sim}$, and $D_{QoE}S_{QoE}$. For the prioritized dimension variant the priority order spatial, temporal, quality was used. This way, the configuration describes the default mechanisms that was used in the system described by [She10]. To investigate the differences between the five configurations, a number of session and SVC video quality metrics are used.



(a) Average total stalling time per peer

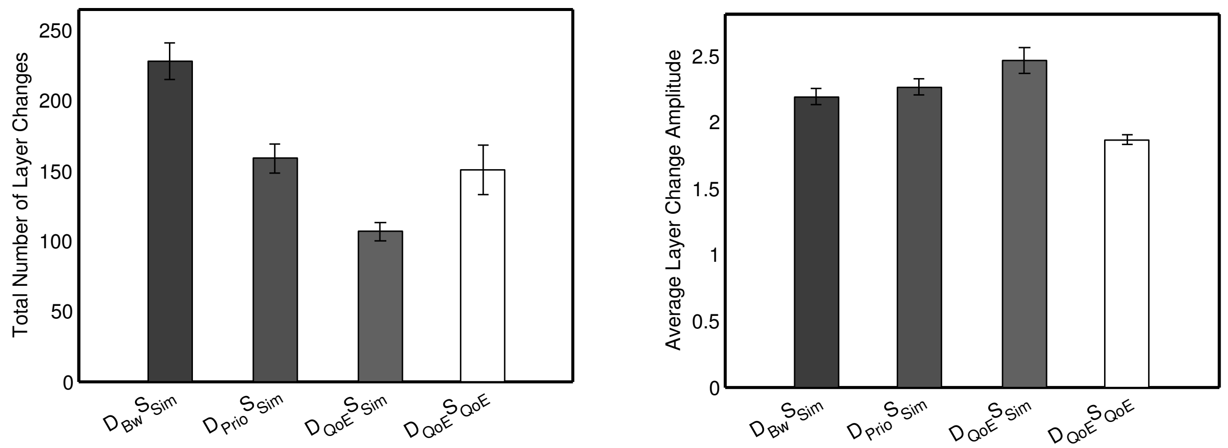
(b) Total number of stalling events over all peers

(c) Total number of layer changes over all peers

Figure 5.5: Scenario 3: Comparison of the session and SVC video quality for different adaptation variants.

Analysis

The main results of the simulative evaluation of this scenario are shown in Figure 5.5 and Figure 5.7. For the first session quality metric, the average total stalling time shown in Figure 5.5a, a significant difference between the $D_{Sim}S_{Sim}$ and all other configurations can be observable. It shows a total stalling time between 25 and 30 seconds, while all other variants reach values below 15 seconds. The same trend can be observable for the total number of stalling events, shown in Figure 5.5b, where the difference is even higher. $D_{Sim}S_{Sim}$ shows about 800 occurred stalling events for the whole system, whereas $D_{Bw}S_{Sim}$, the mechanism with the second highest result, reaches a count of about 120. All others show less than 50 stalling events over the whole streaming process. Besides the slightly increased values of $D_{Bw}S_{Sim}$, all other variants show almost the same performance for these two metrics. The same is true for the third metric, the total number of layer changes, shown in Figure 5.5c. The $D_{Sim}S_{Sim}$ mechanism accounts for about 1700 to 1800 adaptations, whereas the others show values of less than 300 changes. The difference in this metric might be explained, apart from the large number of stalling events, with the adaptation characteristics of the mechanism itself. While other mechanisms may trigger layer increases with large change amplitudes to directly meet the peer's available resources, the $D_{Sim}S_{Sim}$ variant always increases the selected layer in steps of size one.



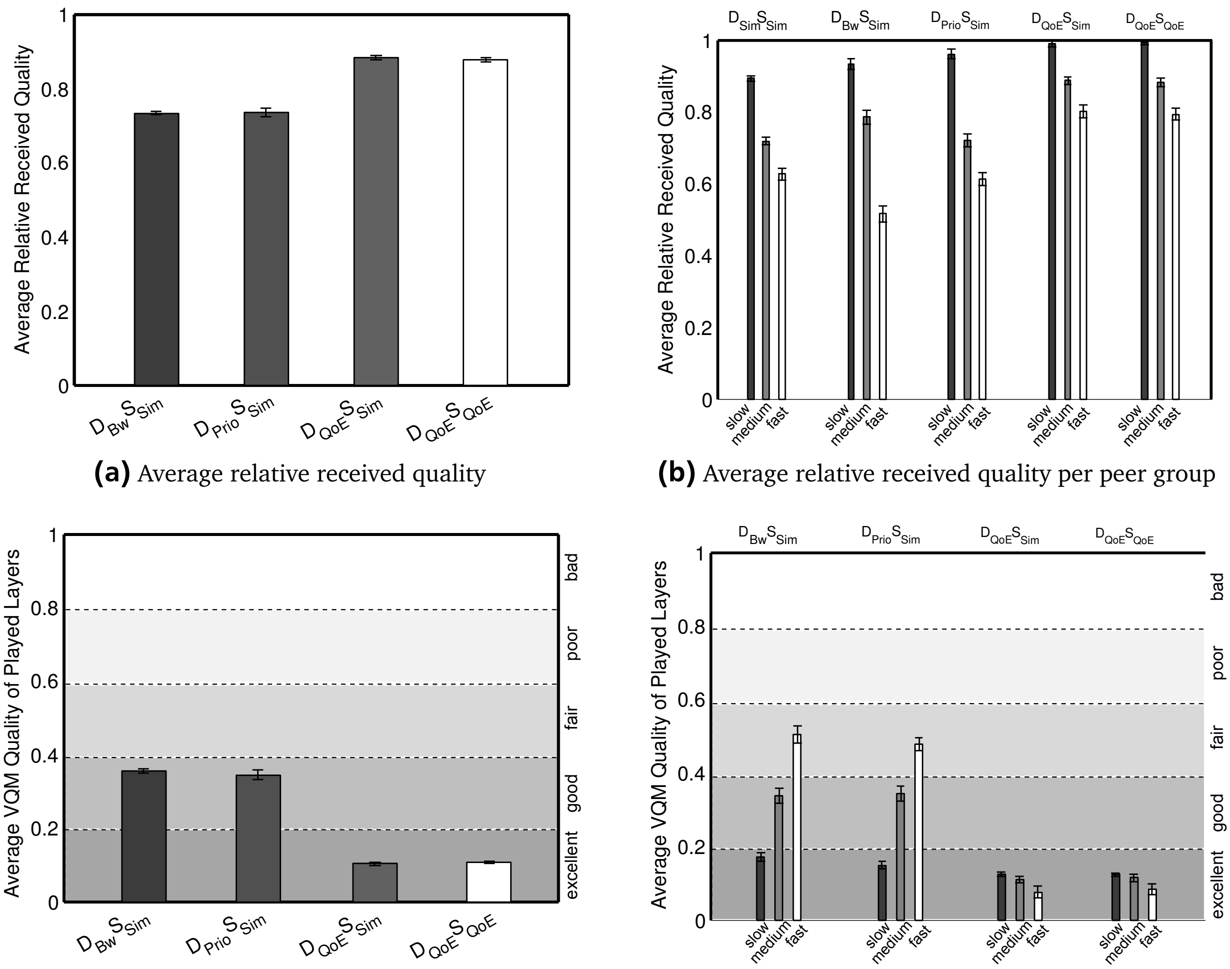
(a) Total number of layer changes over all peers

(b) Average Layer Change Amplitude

Figure 5.6: Scenario 3: Layer change characteristics, including the total number of SVC layer changes, and the average change amplitude.

Due to the bad performance of the configuration $D_{Sim}S_{Sim}$, this variant was excluded from the further plots of this scenario to gain more insight to the differences of the other mechanisms. Therefore, in Figure 5.6 the number of layer changes as well as the average change amplitude are shown again, while excluding $D_{Sim}S_{Sim}$. It is notable that the $D_{QoE}S_{Sim}$ variant shows the highest average change amplitude of about 2.5 and that this could be reduced to a value just below 2 by applying the layer switching mechanism. While the total number of layer changes increased to some extent due to the switching, the reduction of the average step size during the adaptation process is exactly what was intended by the introduction of layer switching. As stated earlier, according to [ZKSS03] the amplitude and frequency of layer changes should be kept as small as possible. By applying the layer switching mechanisms, the amplitude could be reduced at the cost of an increased number of changes. This shows that there is a trade-off between these two session quality aspects.

Further SVC video quality metrics are shown in Figure 5.7, while presenting the average values per configuration and per peer groups. For the average relative received quality, shown in Figure 5.7a and Figure 5.7b, two things are observable. The first is that the non-QoE-aware mechanisms are able to provide an average received quality of about 70%, while the QoE-aware mechanisms show about 90%.



(a) Average relative received quality

(b) Average relative received quality per peer group

(c) Average played VQM quality

(d) Average played VQM quality per peer group

Figure 5.7: Scenario 3: Comparison of the SVC video quality for different adaptation variants.

The second observation that can be made is that for all mechanisms the peers of the slow group have a higher relative received quality than the medium and fast peers. Furthermore, the medium peers have a higher relative quality than the fast peers, which seems rather surprising. Recall that this quality measurement is defined differently for each of the peer groups. Therefore, it depends on the maximum layer determined by the IQA process for each group. Retrieving the maximum layer in the fast group, requires a relatively much higher bit rate than for the slow group. While the peers of the fast group also have a higher bandwidth, they nevertheless depend on the same connection and download management of the overlay.

In contrast, the measured average VQM quality, shown in Figure 5.7c and Figure 5.7d, provides another angle of view on the SVC video quality measurements. First of all, it shows that the non-QoE-aware adaptation variants' results, with a value of just below 0.4 on the VQM scale, are significantly higher than the ones of the QoE-aware mechanisms, which show average VQM values around 0.1. Due to the definition of the VQM, which defines 0.0 as the best and 1.0 as the worst quality rating, this means the QoE-aware mechanisms are able to provide the video with a better estimated perceived quality for the user. Furthermore, the results per group, depicted in Figure 5.7d, show for the non-QoE-aware mechanism that faster peers, although having more resources, receive a lower quality than the slow peers. The relations that were observable for the relative received quality seems to be directly reflected in the estimated perceived quality. The QoE-aware mechanisms, on the other hand, show an inversion of this relation, although the relative received quality showed similar relations between the peer groups. The group of fast peers, in this case, is able to retrieve the content with the best estimated perceived quality, followed by the medium peers and then the slow peers. The latter show the lowest estimated perceived quality. Therefore, the objective to consider the QoE characteristics during the adaptation process showed a positive effect. It is worth noting that user studies have to be performed in order to check whether the proposed mechanisms improve the performance for real users. Nevertheless, the results show a great potential for QoE improvements by considering the content's characteristics during the adaptation process.

The discussed results do not show any further noticeable differences between the QoE-aware adaptation mechanisms with and without applied layer switching mechanism. Nevertheless, the results show that the introduction of layer switching does allow to reduce the layer change amplitude, which, according to Zink et al. [ZKSS03], has an impact on the perceived quality by real users. Since this impact on the QoE is not directly reflected by the metrics that were used in these studies, it would be interesting to define a single metric that allows to reflect more characteristics of the streaming process that have an impact on the estimated perceived quality, besides the pure video quality.

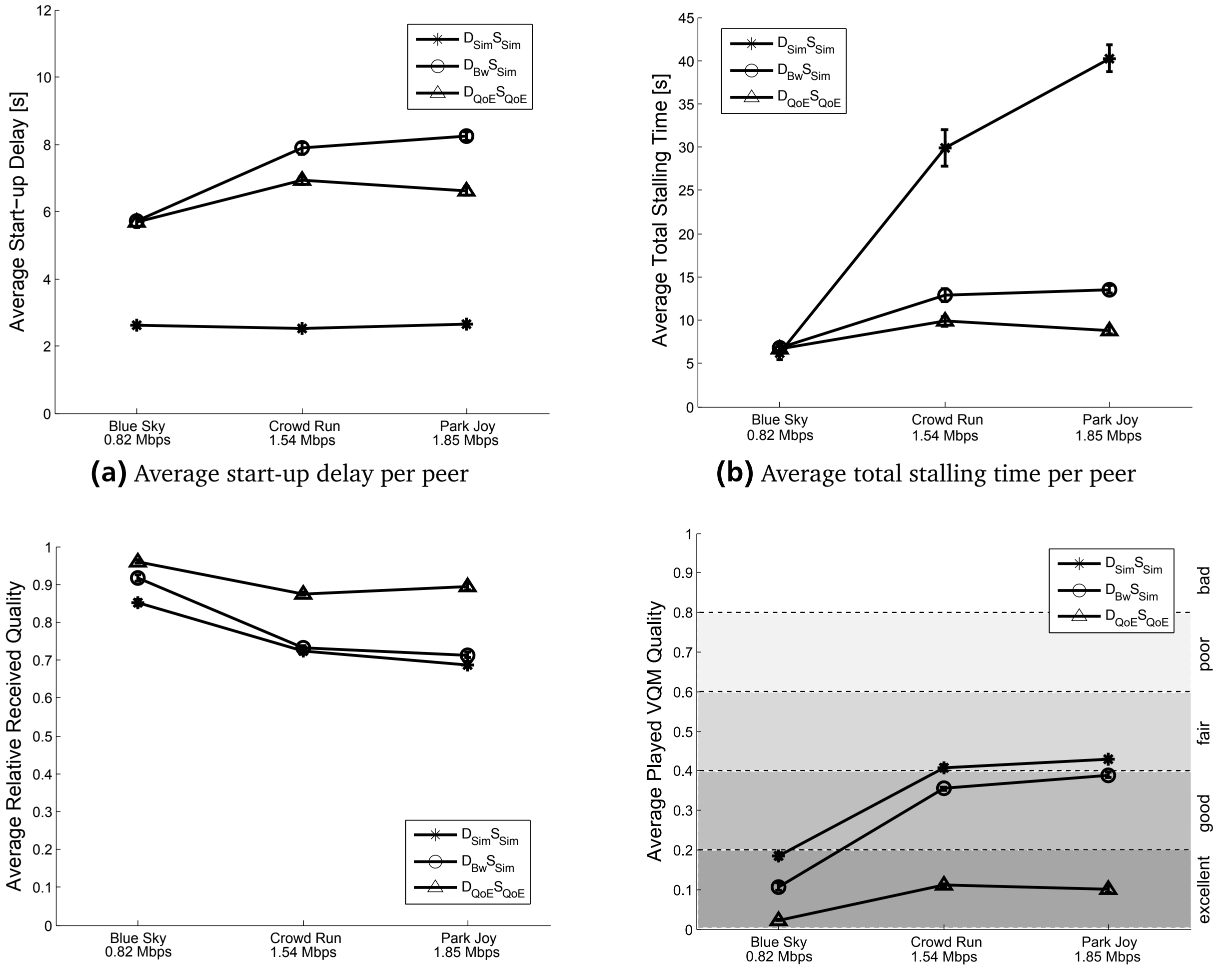
5.4.4 Scenario 4: Videos

This last scenario evalautes the impact that the concrete modeled video has on the performance of the adaptation mechanism. Therefore, all three test videos, which were introduced at the beginning of this section, are used and compared, namely: *Blue Sky*, *Crowd Run*, and *Park Joy*. As these videos strongly vary in their bit rates, although comprising the same maximum spatial and temporal properties, this scenario shows the evaluation of different system capacities. Nevertheless, this scenario helps to understand the impact of the content on the streaming system's session and video quality. Apart from the bit rate, the videos also differ in their VQM ratings, used by the QoE-aware adaptation mechanisms. The following comparison was limited to the configurations $D_{Sim}S_{Sim}$, $D_{Bw}S_{Sim}$, and $D_{QoE}S_{QoE}$ to show basic tendencies.

Analysis

In Figure 5.8, a sub-set of the measurements for the session and SVC video quality is presented. The start-up delay, in Figure 5.8a, shows for the $D_{Sim}S_{Sim}$ configuration a stable value just below 3 seconds, whereas for $D_{Bw}S_{Sim}$ a steady increase from a value below 6 seconds for the lowest bit rate to about 8 seconds for the highest bit rate video can be observable. For the $D_{QoE}S_{QoE}$ configuration, first an increase from a value below 6 seconds to almost 7 seconds for the second video, and a slight decrease of about half a second for the last video can be observable. The stable delay for the $D_{Sim}S_{Sim}$ configuration seems to be a direct result of the properties of the videos' base layers. Although the maximum SVC layers of the videos greatly differ in their bit rates, the base layers all show a rate in the range of 80 to 100 KBps (see Table 5.2 for details). While the video *Blue Sky* in its full quality has the lowest bit rate, for the base layer it even has the highest rate with 99.502 KBps. This way, the $D_{Sim}S_{Sim}$ configuration is able to provide almost the same start-up delay for all three videos. As the other configurations directly start with the retrieval of higher layers, according to their initial layer choice criterion, they show a higher dependency on the increasing bit rates of the videos. Still, it is notable that also for those videos, depending on the choice of the initial layer, the start-up time can be lower for a video with a higher maximum bit rate than for the one with a lower bit rate, due to the lower layer's bit rate characteristics.

Concerning the total stalling time, shown in Figure 5.8b, as in the case of a reduced system capacity, for the configuration $D_{Sim}S_{Sim}$ a steady and steep increase can be observable. It starts with a value of about 6 seconds for the first video, reaches about 30 seconds for the second video, and about 40 seconds for the last video with the highest maximum bit rate. The other configurations also show an increase from the first to the second video but from a value of only 6 seconds to about 13 seconds for $D_{Bw}S_{Sim}$ and to about 10 seconds for $D_{QoE}S_{QoE}$. From the second to the last video, the value for $D_{QoE}S_{QoE}$ is almost stable, and $D_{QoE}S_{QoE}$ even shows a slight decrease by about 2 seconds. This again shows the poor performance of



(a) Average start-up delay per peer

(b) Average total stalling time per peer

(c) Average relative received quality

(d) Average played VQM quality

Figure 5.8: Scenario 4: Comparison of session and SVC video quality for different videos.

the $D_{Sim}S_{Sim}$ configuration, which cannot provide a stable level of service quality for the different videos. $D_{Bw}S_{Sim}$ and $D_{QoE}S_{QoE}$, in contrast, show small changes in the stalling behavior. $D_{QoE}S_{QoE}$ shows the best results and manages to provide a low total stalling time in the range of about 6 to 10 seconds.

The measurements for the SVC video quality are shown in Figure 5.8c and Figure 5.8d. For the average relative received quality, both the configurations $D_{Sim}S_{Sim}$ and $D_{Bw}S_{Sim}$ show a steady decrease in quality. Starting from a value about 90% for $D_{Bw}S_{Sim}$ and 85% for $D_{Sim}S_{Sim}$, they both reach almost the same level just above 70% for the second and around 70% for the last video. $D_{QoE}S_{QoE}$, in contrast, starts with a value of about 95%, falling to about 88% for the second, and rising back to about 90% for the last video. The same tendencies are observable for the average played VQM quality. $D_{Sim}S_{Sim}$ and $D_{Bw}S_{Sim}$ show a steady increase in the VQM value, implying a decrease in estimated perceived quality. $D_{Bw}S_{Sim}$ shows a VQM value of about 0.19 for the first, a value of about 0.40 for the second, and 0.42 for the last video. This means, it starts with an excellent estimated perceived quality on the MOS scale, is reduced to a good quality for the second video and exhibits a fair quality for the third video. $D_{Bw}S_{Sim}$ starts with a value of about 0.10, rises to about 0.35% for the second, and to about 0.38 for the last video. This indicates an excellent estimated perceived quality for the first video and a good quality for the other two. $D_{QoE}S_{QoE}$, in contrast, starts with the overall lowest observed value of about 0.02, rises to about 0.10, and slightly falls back to a VQM value just below this for the last video. This indicates an excellent estimated perceived quality on the MOS scale for all three videos. In sum, the $D_{QoE}S_{QoE}$ configuration, again, shows the best performance with an average relative received quality between 88% and 95% and an excellent estimated perceived quality with a VQM value between 0.02 and 0.10.

To sum these results up, it can be observed that the $D_{Sim}S_{Sim}$ strategy exhibits a low startup delay while the QoE-aware $D_{QoE}S_{QoE}$ showed the best performance in terms of both high session and SVC video qualities.

6 Conclusion

In this work, the potentials of using information about the perceived video quality to realize quality adaptation mechanisms for P2P multi-layer video streaming were investigated. For this purpose, the inherent properties and challenges of P2P streaming systems were described, multi-layer video coding and *Scalable Video Coding* (SVC), as well as the area of video quality assessment and *Quality-of-Experience* (QoE) were introduced. Subsequently, related works in these three areas were described and discussed.

As the main contribution of this work, the design of a QoE-aware quality adaptation mechanism for P2P-based multi-layer video streaming was presented and an implementation in the context of a Video-on-Demand (VoD) P2P streaming system was described. For the adaptation mechanisms, variants with different adaptation granularity were considered. These include both a QoE-aware *Initial Quality Adaptation* (IQA) as well as a QoE-aware *Progressive Quality Adaptation* (PQA) step. The PQA mechanism extended the idea of an already existing progressive adaptation mechanism and introduced two new sub-steps to it, the *Layer Decision* and the *Layer Switching*. While the first is responsible for deriving an adaptation target layer that maximizes the perceived quality, the second was introduced to allow for a smooth transition between the two layers by defining an adaptation path.

Furthermore, for the evaluation a possible concrete realization of the generalized adaptation mechanism was defined by applying the concrete state-of-the-art objective QoE metric VQM and the multi-layer video codec SVC. An implementation of this approach was used for a simulative evaluation as part of a VoD P2P streaming system. The evaluation comprised different scenarios to investigate the performance of the proposed mechanisms in the context of several varying conditions.

The results of the evaluation showed a great potential for the applicability of QoE characteristics during the quality adaptation process in P2P video streaming. In particular, it showed that by applying the proposed QoE-aware adaptation mechanism a content provider could reduce the system capacity in terms of streaming server bandwidth by up to 60% in comparison to the used non-QoE-aware mechanism, while still guaranteeing a high as well as stable session and SVC video quality for the test videos applied. In general, the QoE-aware adaptation mechanism showed advantages over the non-QoE-aware mechanisms in terms of a lower number of total stalling events by up to 50%, as well as the relative received quality of the single peers by up to 20%. Besides, it could be shown that the proposed mechanism succeeds in maximizing the objectively estimated perceived quality, which is the target metric of the mechanism. By additionally using *Layer Switching* mechanisms for smooth transitions, the average layer change amplitude can be reduced, at the cost of an increased total number of changes. For the latter result it is assumed that subjective studies have to be conducted to further evaluate the two parameters' impact on the perceived quality by human users. The results then can be applied to decide on how these parameters have to be weighted to maximize the QoE during streaming.

Additionally, the evaluation showed that one of the used non-QoE-aware system configurations that was used as reference, namely $D_{Sim}S_{Sim}$, is not enough as adaptation mechanism. The only advantage of this configuration is a very low and stable start-up delay, as a result of the initial layer selection mechanism that always selects the base layer at the beginning of a peer's streaming process, regardless of this peer's resources. As this layer has the lowest bit rate of the stream, it allows for a fast content retrieval and this way for a short start-up delay.

Overall, the results show that considering QoE characteristics is highly valuable, since pure maximization of QoS metrics (like throughput) does not directly account for the properties of human perception. Without considering QoE characteristics, layers with lower perceived quality might be preferred over layers with higher perceived quality, since the former have a higher bit rate. The results of the evaluation show that this is in fact the case and that by selecting lower bit rate layers with a higher perceived quality, great opportunities for a reduction of provisioned streaming server capacities arise.

7 Appendix

In this chapter, for the sake of completeness, examples for the configurations and the QoE table files that were used for the evaluation of the adaptation mechanisms are presented.

7.1 Basic Configuration Files

In the following, relevant configuration files are presented. These are the main XML configuration file and the default overlay properties file.

Main Configuration File

Listing 7.1 shows the content of an example main configuration file used for the simulative evaluation with *PeerfactSim.KOM*.

```xml
<?xml version='1.0' encoding='utf-8'?>
<Configuration>
  <Default>
    <Variable name="seed" value="0"/>
    <Variable name="size" value="110"/>
    <Variable name="finishTime" value="300m"/>
    <Variable name="gnpDataFile" value="data/measured_data.xml"/>
    <Variable name="description" value="STREAMING_GNP"/>
    <Variable name="streamingPropertiesFile"
        value="config/Streaming/Adaptation_Mechanisms/Scenario0_DbwSsim.properties"/>
  </Default>

  <SimulatorCore seed="$seed" finishAt="$finishTime" statusInterval="5m" />

  <NetLayer
      class="de.tud.kom.p2psim.impl.network.optimizedInitial.OptimizedNetLayerFactory"
      gnpFile="$gnpDataFile"/>

  <TransLayer class="de.tud.kom.p2psim.impl.transport.OptimizedTransLayerFactory"/>

  <Overlay class="de.tud.kom.p2psim.streaming.overlay.StreamingOverlayFactory"
      propertyFile="$streamingPropertiesFile"/>

  <ApplicationLayer class="de.tud.kom.p2psim.streaming.application.StreamingAppFactory"/>

  <UserLayer class="de.tud.kom.p2psim.streaming.user.StreamingUserFactory"/>

  <Monitor class="de.tud.kom.p2psim.impl.common.DefaultMonitor" start="0"
      stop="$finishTime">
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.LayerAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.StallingAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.ReceivedQualityAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.ThroughputAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.NodeStateAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.EnhancedThroughputAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.MetaDataAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.ReceivedQoEAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.RuntimePerformanceAnalyzer" />
```

```
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.VideoBitrateAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.AdaptationAnalyzer" />
    <Analyzer
        class="de.tud.kom.p2psim.streaming.analyzer.BufferBlockAvailabilityAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.UpThroughputOverTimeAnalyzer" />
    <Analyzer class="de.tud.kom.p2psim.streaming.analyzer.DownThroughputOverTimeAnalyzer"
        />
  </Monitor>

  <HostBuilder class="de.tud.kom.p2psim.streaming.general.StreamingHostBuilder"
      experimentSize="$size">

      <!-- tracker -->
    <Host groupID="Portugal">
      <NetLayer upBandwidth="50000Kbps" downBandwidth="100000Kbps" />
      <TransLayer />
      <Overlay isTracker="true" port="2" />
      <ApplicationLayer isTracker="true" />
      <Properties enableChurn="false" />
    </Host>

    <!-- seeder -->
    <Group size="9" groupID="Spain" isSeeder="true">
      <NetLayer upBandwidth="25000Kbps" downBandwidth="50000Kbps" />
      <TransLayer />
      <Overlay isTracker="false" port="1" screen="1"
        computingPower="2000" />
      <ApplicationLayer isTracker="false" />
      <UserLayer />
      <Properties enableChurn="false" />
    </Group>

    <!-- leecher -->
    <Group size="20" groupID="Switzerland">
      <NetLayer upBandwidth="500Kbps" downBandwidth="2000Kbps" />
      <TransLayer />
      <Overlay isTracker="false" port="3" screen="0"
        computingPower="700" />
      <ApplicationLayer isTracker="false" />
      <UserLayer />
      <Properties enableChurn="false" />
    </Group>

    <!--leecher-->
    <Group size="50" groupID="Germany">
      <NetLayer upBandwidth="3000Kbps" downBandwidth="8000Kbps" />
      <TransLayer />
      <Overlay isTracker="false" port="3" screen="2"
        computingPower="800" />
      <ApplicationLayer isTracker="false" />
      <UserLayer />
      <Properties enableChurn="false" />
    </Group>

    <!-- leecher-->
    <Group size="30" groupID="France">
      <NetLayer upBandwidth="8000Kbps" downBandwidth="16000Kbps" />
      <TransLayer />
      <Overlay isTracker="false" port="3" screen="3"
        computingPower="900" />
```

```
      <ApplicationLayer isTracker="false" />
      <UserLayer />
      <Properties enableChurn="false" />
    </Group>

    </HostBuilder>

  <Workload class="de.tud.kom.p2psim.streaming.user.StreamingWorkload"
      modelType="cubeModel" seeder="Spain" />

  <Scenario class="de.tud.kom.p2psim.streaming.user.StreamingScenario" />

</Configuration>
```

Listing 7.1: An example for the main configuration file with the basic setting used for the evaluations.

Properties file

Listing 7.2 shows an excerpt of the streaming overlay's properties file. It allows to specify the most important streaming overlay parameters for a simulation and is used in addition to the main XML configuration file, which mainly comprises scenario specific settings. The used properties file is specified in the XML configuration file as property of the streaming overlay's XML tag definition.

```
# General
VIDEO_BITRATE = 1000
VIDEO_LENGTH = 300
VIDEO_NAME = 5minVideo

USE_LAYER_RATES_OF_QOE_FILE = true
QOE_AWARE_ADAPTION_TABLE_FILE =
    config/Streaming/QoE_Aware_Adaptation/QoE_table_Crowd-Run.dat

# Inital Quality Adaptation (IQA)
IQA_ENABLED = true

# The strategy to select initial layer
# 0 = Default (= upper band layer)
# 1 = Max. bandwidth utilization
# 2 = Max. QoE rating
# 3 = Base layer
INITIAL_LAYER_CHOICE = 0

# Progressive Quality Adaptation (PQA)
PQA_ENABLED = true
PQA_INTERVAL = 10

USE_NET_STAT_ADAPT = true

# Layer Decision
# 0 = Max. QoE Rating
# 1 = Max. Bandwidth Utilization
# 2 = Simple Layer Decision
# 3 = Prioritized Dimensions
QOE_AWARE_ADAPTATION_LAYER_DECISION_ALGORITHM = 0
```

```
# Layer Switching
# 0 = Prioritized Dimensions
# 1 = Min. QoE variation Dijkstra
# 2 = Min. Bandwidth Variation Dijkstra
# 3 = Simple Layer Switching (= Disable Layer Switching)
QOE_AWARE_ADAPTATION_LAYER_SWITCHING_ALGORITHM = 1

# The priority to be used for prioritized dimensions layer decision and switching
# 0 = d > t > q
# 1 = t > d > q
# 2 = t > q > d
PRIORITIZED_DIMENSIONS_PRIORITY = 0

QOE_AWARE_ADAPTATION_LAYER_SWITCHING_ENABLED = true
QOE_AWARE_ADAPTATION_LAYER_SWITCHING_USE_IF_STALLED = true
QOE_AWARE_ADAPTATION_LAYER_SWITCHING_MINIMAL_TIME_PER_STEP = 4


# Time after which the switching has to be finished [seconds]
LAYER_SWITCHING_SPEED = 8

SWITCH_TO_BL_ON_STALL_DISABLED = false

# Ratio of throughput to be used as upper limit adaptation for adaptations
BITRATE_ADAPTATION_FACTOR = 0.9

# The time window used to compute a peer's average throughput
THROUGHPUT_ESTIMATION_INTERVAL = 20

# Scenario definition
#0 = Simple random arrival with the configured mean arrival time
#1 = Double flash crowd scenario with the configured parameters.
SCENARIO = 1
MEAN_ARRIVAL_TIME = 90s
FIRST_JOIN_PERIOD = 0s-300s
SECOND_JOIN_PERIOD = 450s-750s
PEER_DISTRIBUTION_BALANCE = 0.5


#Peer Leaving
LEAVE_MAX_DELAY = 5
LEAVE_IMMEDIATELY_PROBABILITY = 0.4
LEAVE_IF_STALLING_TOO_LONG = false
LEAVE_TRESHOLD_RELATIVE_TO_MOVIE_LENGTH = 1.0
LEAVE_INITIAL_TRESHOLD_RELATIVE_TO_MOVIE_LENGTH = 2.0
```

Listing 7.2: Excerpt of a properties file used to configure most of the streaming overlay's parameters.

7.2 Test Video Data

Listing 7.3, Listing 7.4, and Listing 7.5 show the content of the three different QoE table files that were used for the simulative evaluation of the adaptation mechanisms with *PeerfactSim.KOM*. The detailed data were provided by Zinner et al. who used this data for their investigations in [ZHAH10]. Details on the process of deriving these values are descriped in their work.

```
# Spatial level (d)
# Temporal level (t)
# SNR level (q)
# Data rate (Mbyte/s)
# VQM value
0 0 0 0.099502  0.7080
1 0 0 0.237859  0.6980
2 0 0 0.344678  0.6991
3 0 0 0.483708  0.7017
0 1 0 0.116962  0.5987
1 1 0 0.278904  0.5755
2 1 0 0.405544  0.5699
3 1 0 0.571487  0.5720
0 2 0 0.133664  0.4173
1 2 0 0.319317  0.3905
2 2 0 0.467042  0.3859
3 2 0 0.662858  0.3886
0 3 0 0.148941  0.2617
1 3 0 0.356181  0.2279
2 3 0 0.521065  0.2230
3 3 0 0.742194  0.2253
0 4 0 0.164336  0.0557
1 4 0 0.394226  0.0188
2 4 0 0.574822  0.0089
3 4 0 0.820000  0.0000
```

Listing 7.3: QoE table for the video *Blue Sky*.

```
# Spatial level (d)
# Temporal level (t)
# SNR level (q)
# Data rate (Mbyte/s)
# VQM value
0 0 0 0.083343  0.7838
1 0 0 0.196833  0.7635
2 0 0 0.326427  0.7580
3 0 0 0.508360  0.7605
0 1 0 0.124848  0.6141
1 1 0 0.295807  0.5816
2 1 0 0.486092  0.5734
3 1 0 0.749941  0.5762
0 2 0 0.174872  0.4276
1 2 0 0.423468  0.3874
2 2 0 0.688186  0.3803
3 2 0 1.046318  0.3846
0 3 0 0.216091  0.2345
1 3 0 0.534859  0.1730
2 3 0 0.881488  0.1635
3 3 0 1.333625  0.1676
0 4 0 0.235070  0.1170
1 4 0 0.591817  0.0318
2 4 0 1.003290  0.0123
3 4 0 1.540000  0.0000
```

Listing 7.4: QoE table for the video *Crowd Run*.

```
# Spatial level (d)
# Temporal level (t)
# SNR level (q)
# Data rate (Mbyte/s)
# VQM value
0 0 0 0.089190  0.7204
1 0 0 0.195647  0.6977
2 0 0 0.338438  0.6896
3 0 0 0.568799  0.6913
0 1 0 0.134219  0.5629
1 1 0 0.298488  0.5298
2 1 0 0.528358  0.5213
3 1 0 0.890658  0.5230
0 2 0 0.180994  0.4292
1 2 0 0.418088  0.3900
2 2 0 0.778516  0.3863
3 2 0 1.328312  0.3825
0 3 0 0.197295  0.2924
1 3 0 0.473345  0.2418
2 3 0 0.956511  0.2317
3 3 0 1.703705  0.2350
0 4 0 0.202510  0.1064
1 4 0 0.487983  0.0306
2 4 0 1.004883  0.0126
3 4 0 1.850000  0.0000
```

Listing 7.5: QoE table for the video *Park Joy*.

List of Figures

List of Tables

Bibliography

[APKS09] O. Abboud, K. Pussep, A. Kovacevic, and R. Steinmetz. Quality Adaptive Peer-to-Peer Streaming Using Scalable Video Coding. In *IFIP/IEEE MMNS*, 2009.

[APM+10] O. Abboud, K. Pussep, M. Mueller, A. Kovacevic, and R. Steinmetz. Advanced Prefetching and Upload Strategies for P2P Video-on-Demand. In *ACM AVSTP2P*, 2010.

[AZPS11] O. Abboud, T. Zinner, K. Pussep, and R. Steinmetz. On the Impact of Quality Adaptation in SVC-based P2P Video-on-Demand Systems. In *ACM MMSys*, 2011.

[BSWG07] P. Baccichet, T. Schierl, T. Wiegand, and B. Girod. Low-delay Peer-to-Peer Streaming using Scalable Video Coding. In *IEEE PV workshop*, 2007.

[Cis11] Cisco Systems Inc. Cisco VNI: Forecast and Methodology, 2010-2015. `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf`, 2011. [Accessed Nov. 12, 2011].

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, editors. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[CN03] Y. Cui and K. Nahrstedt. Layered Peer-to-Peer Streaming. In *ACM NOSSDAV*, 2003.

[EZ07] U. Engelke and H.-J. Zepernick. Perceptual-based Quality Metrics for Image and Video Services: A Survey. In *EuroNGI NGI*, 2007.

[FLM+10] R. Fortuna, E. Leonardi, M. Mellia, M. Meo, and S. Traverso. QoE in Pull Based P2P-TV Systems: Overlay Topology Design Tradeoffs. In *IEEE P2P*, 2010.

[GHP08] J. Gustafsson, G. Heikkila, and M. Pettersson. Measuring Multimedia Quality in Mobile Networks with an Objective Parametric Model. In *IEEE ICIP*, 2008.

[ITU10] International Telecommunication Union. Recommendation ITU-T H.264 : Advanced Video Coding for Generic Audiovisual Services. `http://www.itu.int/rec/T-REC-H.264`, 2010. [Accessed Nov. 12, 2011].

[KSNR09] C. S. Kim, H. Sohn, W. D. Neve, and Y. M. Ro. An Objective Perceptual Quality-Based ADTE for Adapting Mobile SVC Video Content. *IEICE Trans. on Information & Systems*, E92-D(1):pp. 93–96, 2009.

[KTP+10] S. Kaune, G. Tyson, K. Pussep, A. Mauthe, and R. Steinmetz. The Seeder Promotion Problem: Measurements, Analysis and Solution Space. In *IEEE ICCCN*, 2010.

[LSE11] J. Lee, F. De Simone, and T. Ebrahimi. Subjective Quality Evaluation via Paired Comparison: Application to Scalable Video Coding. *IEEE Trans. on Multimedia*, 13(5):pp. 882–893, 2011.

[MA08] M. Mushtaq and T. Ahmed. Smooth Video Delivery for SVC based Media Streaming over P2P Networks. In *IEEE CCNC*, 2008.

[MEL10] V. Menkovski, G. Exarchakos, and A. Liotta. Machine Learning Approach for Quality of Experience Aware Networks. In *IEEE INCoS*, 2010.

[MH10] K. Mokhtarian and M. Hefeeda. Analysis of peer-assisted Video-on-Demand Systems with Scalable Video Streams. In *ACM MMSys*, 2010.

[Mu09] M. Mu. An Interview with Video Quality Experts. *ACM SIGMultimedia Records*, 1(4):pp. 4–13, 2009.

[NEE+11] P. Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen. Flicker Effects in Adaptive Video Streaming to Handheld Devices. In *ACM Multimedia*, 2011.

[OZPH10] S. Oechsner, T. Zinner, J. Prokopetz, and T. Hoßfeld. Supporting Scalable Video Codecs in a P2P Video-on-Demand Streaming System. In *ITC Specialist Seminar on Multimedia Applications - Traffic, Performance and QoE*, 2010.

[PW04] M. Pinson and S. Wolf. A New Standardized Method for Objectively Measuring Video Quality. *IEEE Trans. on Broadcasting*, 50(3):pp. 312–322, 2004.

[San11] Sandvine Inc. Fall 2011 Internet Phenomena Report. `http://www.sandvine.com/downloads/documents/10-26-2011_phenomena/Sandvine%20Global%20Internet%20Phenomena%20Report%20-%20Fall%202011.pdf`, 2011. [Accessed Nov. 12, 2011].

[She10] Y. Sheng. Implementation and Evaluation of Advanced Algorithms for Scalable P2P Video Streaming (KOM-D-383). Master's thesis, Technische Universität Darmstadt, 2010.

[SMW07] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE TCSVT*, 17(9):1103–1120, 2007.

[SW08] H. Schwarz and M. Wien. The Scalable Video Coding Extension of the H.264/AVC Standard [Standards in a Nutshell]. *IEEE Signal Processing Magazine*, 25(2):135–141, 2008.

[VQEG03] Video Quality Experts Group. Final Report on the Validation of Objective Models of Video Quality Assessment, FR-TV Phase II. `http://www.its.bldrdoc.gov/vqeg/projects/frtv_phaseII/`, 2003. [Accessed Nov. 12, 2011].

[WBSS04] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. on Image Processing*, 13(4):600–612, 2004.

[Win05] S. Winkler. *Digital Video Quality: Vision Models and Metrics*. Wiley, 1st edition, 2005.

[Win07] S. Winkler. Video Quality and Beyond. In *EURASIP EUSIPCO*, 2007.

[WP07] S. Wolf and M. Pinson. Application of the NTIA General Video Quality Metric (VQM) to HDTV Quality Monitoring. In *VPQM*, 2007.

[XSGZ09] X. Xiao, Y. Shi, Y. Gao, and Q. Zhang. LayerP2P: A New Data Scheduling Approach for Layered Streaming in Heterogeneous Networks. In *IEEE INFOCOM*, 2009.

[ZAH+10] T. Zinner, O. Abboud, O. Hohlfeld, T. Hoßfeld, and P. Tran-Gia. Towards QoE Management for Scalable Video Streaming. In *ITC Specialist Seminar on Multimedia Applications - Traffic, Performance and QoE*, 2010.

[ZCL+08] G. Zhai, J. Cai, W. Lin, X. Yang, and W. Zhang. Three Dimensional Scalable Video Adaptation via User-End Perceptual Quality Assessment. *IEEE Trans. on Broadcasting*, 54(3):pp. 719–727, 2008.

[ZHAH10] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hoßfeld. Impact of Frame Rate and Resolution on Objective QoE Metrics. In *IEEE QoMEX workshop*, 2010.

[ZKSS03] M. Zink, O. Künzel, J. Schmitt, and R. Steinmetz. Subjective Impression of Variations in Layer Encoded Videos. In *IEEE IWQoS*, 2003.