



**Julius-Maximilians-Universität Würzburg**  
Department for Computer Science  
Chair of Communication Networks (Informatik III)

# Holistic Evaluation of Novel Adaptation Logics for DASH and SVC

## Leistungsbewertung neuartiger Adaptionslogiken für DASH mit SVC

Master's thesis in Computer Science  
by

**Christian Sieber**



**Julius-Maximilians-Universität Würzburg**  
Department for Computer Science  
Chair of Communication Networks (Informatik III)

# Holistic Evaluation of Novel Adaptation Logics for DASH and SVC

## Leistungsbewertung neuartiger Adaptionslogiken für DASH mit SVC

Master's thesis in Computer Science  
submitted by

**Christian Sieber**

born on the 30th of June 1986 in Ochsenfurt

completed at  
Chair of Communication Networks (Informatik III)  
Julius-Maximilians-Universität Würzburg

Supervisors:

Prof. Dr.-Ing. P. Tran-Gia  
Dr. rer. nat. Thomas Zinner  
Dr. rer. nat. Tobias Hoffeld

Submission:

17. 8. 2013

## **Erklärung**

Ich versichere, die vorliegende Masterarbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur verfasst zu haben. Darüber hinaus versichere ich, die Arbeit bisher oder gleichzeitig keiner anderen Prüfungsbehörde zur Erlangung eines akademischen Grades vorgelegt zu haben.

Würzburg, den 17. 8. 2013

---

(Christian Sieber)

# Contents

<b>Table of Contents</b>	<b>iv</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Background and Related Work</b>	<b>5</b>
2.1. Video Coding with H.264 . . . . .	5
2.1.1. Advanced Video Coding . . . . .	5
2.1.2. Scalable Video Coding . . . . .	6
2.2. MPEG-DASH for Video Streaming . . . . .	7
2.2.1. HTTP Progressive Download . . . . .	8
2.2.2. MPEG - Dynamic Adaptive Streaming over HTTP . . . . .	8
2.2.3. Comparison to Datagram-based Streaming . . . . .	9
2.2.4. Investigated Adaptation Algorithms . . . . .	10
2.2.4.1. TRDA . . . . .	11
2.2.4.2. KLUDCP . . . . .	12
2.2.4.3. Tribler . . . . .	12
2.3. Scalable Video Coding with DASH . . . . .	13
2.4. QoE of Video Streaming . . . . .	14
2.4.1. Temporal / Spatial Information . . . . .	14
2.4.2. Influence Factors of SVC-based Video Streaming . . . . .	15
2.4.2.1. Temporal Impairments . . . . .	15
2.4.2.2. Video Quality . . . . .	15
2.4.2.3. Flicker Effects . . . . .	16
2.4.3. QoE Assessment . . . . .	16
2.4.3.1. Video Quality . . . . .	16
2.4.3.2. Subjective Methods . . . . .	18
2.4.3.3. Crowdsourcing for QoE Assessment . . . . .	19
<b>3. Subjective Studies</b>	<b>20</b>
3.1. Problem Formulation . . . . .	21
3.1.1. Research Questions . . . . .	22
3.1.2. Control Questions . . . . .	22
3.2. Pilot Study in a Laboratory . . . . .	23
3.2.1. Methodology and Lab Setup . . . . .	24
3.2.2. Results of the Pilot Study . . . . .	25
3.3. Methodology for Crowdsourcing QoE Assessments . . . . .	26
3.3.1. Web-based User Interface . . . . .	27
3.3.2. Test Scenes and Quality Levels . . . . .	28
3.3.3. Filtering Unreliable Test Subjects . . . . .	28

3.3.4. Crowdsourcing Campaigns . . . . .	29
3.4. Results of the statistical analysis of the subjective studies . . . . .	30
3.4.1. Demographic of the Crowd . . . . .	31
3.4.2. Perception of Quality Switches . . . . .	31
3.4.3. Impact of Quality Switches on Perceived Video Quality . . . . .	32
3.4.4. Acceptance of Quality Switches . . . . .	34
3.4.5. Identified QoE Influence Factors . . . . .	36
<b>4. An Adaptation Algorithm and Methodology for Objective Evaluation</b>	<b>38</b>
4.1. Bandwidth Independent Efficient Buffering (BIEB) Algorithm . . . . .	39
4.2. Utilized Evaluation Metrics . . . . .	43
4.2.1. Resource-centric Metrics . . . . .	43
4.2.2. User-centric Metrics . . . . .	44
4.3. Content Characteristics . . . . .	44
4.4. Testbed Environment . . . . .	45
4.5. Network Access Characteristics . . . . .	46
4.5.1. Constant Bandwidth Limitation . . . . .	46
4.5.2. Vehicular Mobility . . . . .	47
4.6. Investigated Scenarios . . . . .	47
4.6.1. Vehicular Mobility . . . . .	47
4.6.2. Scalability to Bandwidth . . . . .	48
4.6.3. Fairness for Two Clients . . . . .	48
4.6.4. Fairness for Cross-Traffic . . . . .	48
4.7. Evaluation Framework for HTTP DASH Measurements . . . . .	49
<b>5. Performance Evaluation of DASH Adaptation Algorithms</b>	<b>51</b>
5.1. Evaluation in the Vehicular Mobility Scenario . . . . .	51
5.1.1. Qoe Influence Factors . . . . .	51
5.1.2. Efficiency and Usage of Resources . . . . .	54
5.2. Playback Quality and Switching Scalability . . . . .	55
5.3. Playback Quality and Switching Fairness . . . . .	57
5.4. Bandwidth Fairness for Cross-Traffic . . . . .	59
5.5. Comparison of the Investigated Algorithms . . . . .	60
<b>6. Conclusion and Outlook</b>	<b>63</b>
<b>A. Appendix</b>	<b>65</b>
A.1. Crowd-sourcing Demographic Campaign C1 . . . . .	65
A.2. File Listing . . . . .	66
A.3. Campaigns . . . . .	67
A.4. Web-based Crowdsourcing Interface . . . . .	67
A.5. Crowdsourcing Questionnaire . . . . .	74
A.6. Microworkers.com Campaign Description . . . . .	75
A.7. Crowdsourcing Campaign Introduction . . . . .	75
<b>List of Figures</b>	<b>78</b>

<b>List of Tables</b>	<b>79</b>
<b>Bibliography</b>	<b>80</b>

# 1. Introduction

Streaming of videos has become the major traffic generator in today's Internet and the video traffic share is still increasing. According to Cisco's annual Visual Networking Index report [4], in 2012, 60 % of the global Internet IP traffic was generated by video streaming services. Furthermore, the study predicts further increase to 73 % by 2017. At the same time, advances in the fields of mobile communications and embedded devices lead to a widespread adoption of Internet video enabled mobile and wireless devices (e.g. Smartphones). The report predicts that by 2017, the traffic originating from mobile and wireless devices will exceed the traffic from wired devices and states that mobile video traffic was the source of roughly half of the mobile IP traffic at the end of 2012.

With the increasing importance of Internet video streaming in today's world, video content provider find themselves in a highly competitive market where user expectations are high and customer loyalty depends strongly on the user's satisfaction with the provided service. In particular paying customers expect their viewing experience to be the same across all their viewing devices and independently of their currently utilized Internet access technology. However, providing video streaming services is costly in terms of storage space, required bandwidth and generated traffic. Therefore, content providers face a trade-off between the user perceived Quality of Experience (QoE) [51] and the costs for providing the service.

Today, a variety of transport and application protocols exist for providing video streaming services, but the one utilized depends on the scenario in mind. Video streaming services can be divided up in three categories: Video conferencing, IPTV and Video-on-Demand services. IPTV and video-conferencing have severe real-time constraints and thus utilize mostly datagram-based protocols like the RTP/UDP protocol for the video transmission. Video-on-Demand services in contrast can profit from pre-encoded content, buffers at the end user's device, and mostly utilize TCP-based protocols in combination with progressive streaming for the media delivery.

In recent years, the HTTP protocol on top of the TCP protocol gained widespread popularity as a cost-efficient way to distribute pre-encoded video content to customers via progressive streaming. This is due to the fact that HTTP-based video streaming profits from a well-established infrastructure which was originally implemented to efficiently satisfy the increasing demand for web browsing and file downloads. Large Content Delivery Networks (CDN) are the key components of that distribution infrastructure. CDNs prevent expensive long-haul data traffic and delays by distributing HTTP content to world-wide locations close to the customers. As of 2012, already 53 % of the global video traffic in the Internet originates from Content Delivery Networks and that percentage is expected to increase to 65 % by the year 2017. Furthermore, HTTP media streaming profits from existing HTTP caching infrastructure, ease of NAT and proxy traversal and firewall friendliness.

Video delivery through heterogeneous wired and wireless communications networks is prone to distortions due to insufficient network resources. This is especially true in wireless scenarios, where user mobility and insufficient signal strength can result in a very poor transport service performance (e.g. high packet loss, delays and low and varying bandwidth). A poor performance of the transport in turn may degrade the Quality of Experience as perceived by the user, either due to buffer underruns (i.e. playback interruptions) for TCP-based delivery [36] or image distortions for datagram-based real-time video delivery.

In order to overcome QoE degradations due to insufficient network resources, content providers have to consider adaptive video streaming. One possibility to implement this for HTTP/TCP streaming is by partitioning the content into small segments, encode the segments into different quality levels and provide access to the segments and the quality level details (e.g. resolution, average bitrate). During the streaming session, a client-centric adaptation algorithm can use the supplied details to adapt the playback to the current environment. However, a lack of a common HTTP adaptive streaming standard led to multiple proprietary solutions developed by major Internet companies like Microsoft (Smooth Streaming), Apple (HTTP Live Streaming) and Adobe (HTTP Dynamic Streaming) loosely based on the aforementioned principle. In 2012, the ISO/IEC published the Dynamic Adaptive Streaming over HTTP (MPEG-DASH) standard. As of today, DASH is becoming widely accepted with major companies announcing their support or having already implemented the standard into their products. MPEG-DASH is typically used with single layer codecs like H.264/AVC, but recent publications [55] show that scalable video coding can use the existing HTTP infrastructure more efficiently. Furthermore, the layered approach of scalable video coding extends the adaptation options for the client, since already downloaded segments can be enhanced at a later time.

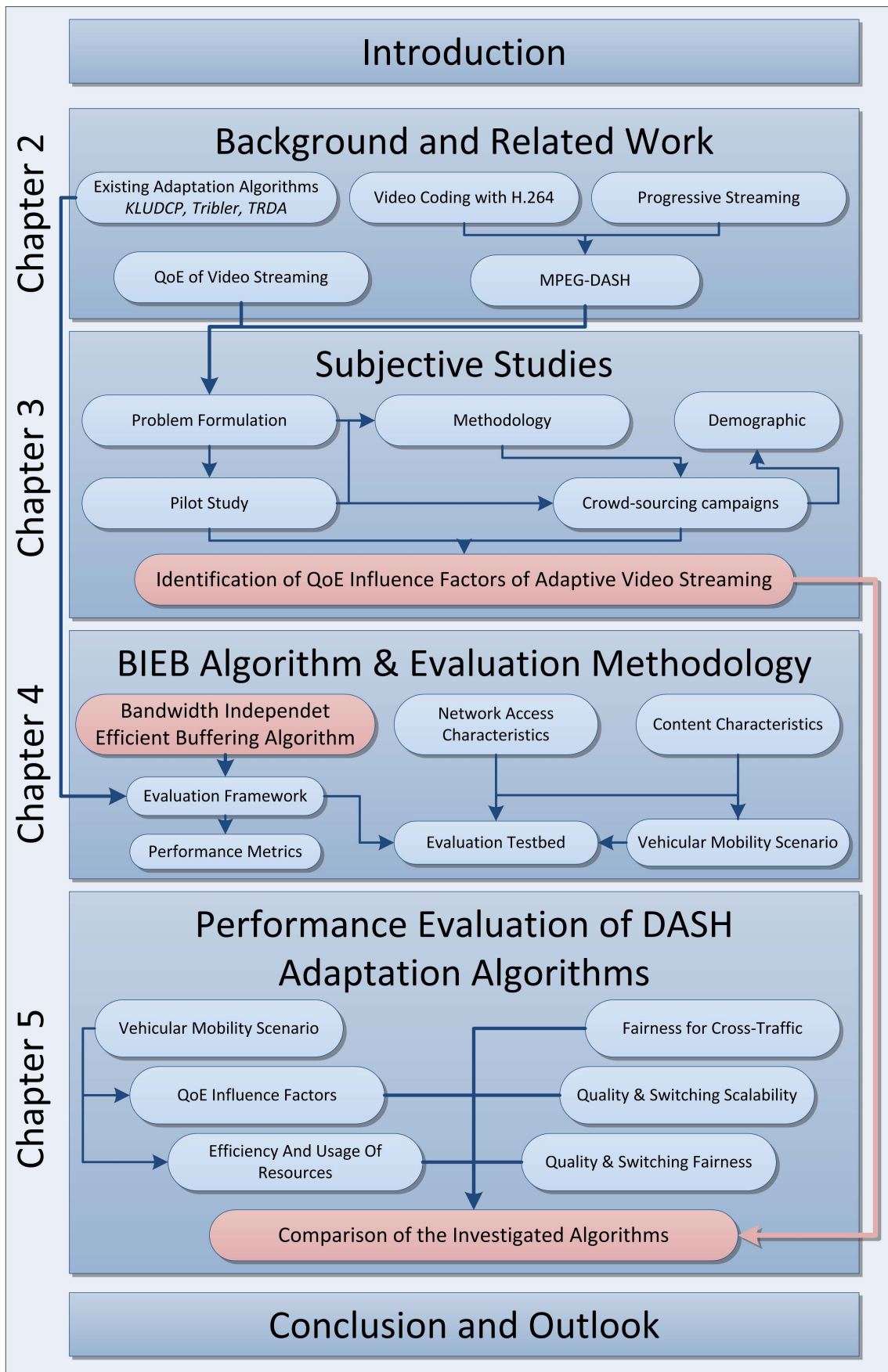
The influence of distortions on the perceived QoE for non-adaptive video streaming are well reviewed and published. For HTTP streaming, the QoE of the user is influenced by the initial delay (i.e. the time the client pre-buffers video data) and the length and frequency of playback interruptions due to a depleted video playback buffer. Studies highlight that even low stalling times and frequencies have a negative impact on the QoE of the user and should therefore be avoided. The first contribution of this thesis is the *identification of QoE influence factors of adaptive video streaming* by the means of crowd-sourcing and a laboratory study.

MPEG-DASH does not specify how to adapt the playback to the available bandwidth and therefore the design of a download/adaptation algorithm is left to the developer of the client logic. The second contribution of this thesis is the *design of a novel user-centric adaptation logic for DASH with SVC*. Other download algorithms for segmented HTTP streaming with single layer and scalable video coding have been published lately. However, there is little information about the behavior of these algorithms regarding the identified QoE-influence factors. The third contribution is a *user-centric performance evaluation of three existing adaptation algorithms and a comparison to the proposed algorithm*. In the performance evaluation we also evaluate the fairness of the algorithms. In one fairness scenario, two clients deploy the same adaptation algorithm and share one Internet connection. For a fair adaptation algorithm, we expect the behavior of the two clients to be identical. In a second



fairness scenario, one client shares the Internet connection with a large HTTP file download and we expect an even bandwidth distribution between the video streaming and the file download. The forth contribution of this thesis is an *evaluation of the behavior of the algorithms in a two-client and HTTP cross traffic scenario*.

The remainder of this thesis is structured as follows. Chapter II gives a brief introduction to video coding with H.264, the HTTP adaptive streaming standard MPEG-DASH, the investigated adaptation algorithms and metrics of Quality of Experience (QoE) for video streaming. Chapter III presents the methodology and results of the subjective studies conducted in the course of this thesis to identify the QoE influence factors of adaptive video streaming. In Chapter IV, we introduce the proposed adaptation algorithm and the methodology of the performance evaluation. Chapter V highlights the results of the performance evaluation and compares the investigated adaptation algorithms. Section VI summarizes the main findings and gives an outlook towards QoE-centric management of DASH with SVC.



Organization and contribution of this thesis

## 2. Background and Related Work

In the following, we give a brief introduction of the relevant technologies used in this thesis and related work published on the specific areas of research. First, we explain the theory of video coding with H.264 and its scalable video coding extension. Second, we introduce the progressive media streaming standard MPEG-DASH and proceed by differentiating progressive HTTP streaming to datagram-based streaming such as RTP/UDP live streaming. Afterwards, we introduce three published DASH adaptation algorithms, two for single layer coded video content and one for scalable video coded content. Next, we point out how progressive video streaming can benefit from the use of scalable video coding compared to single layer coding. In the last section of this chapter we discuss the Quality of Experience as perceived by a user for video streaming in general and in particular for progressive video streaming utilizing scalable video coding. To do so, we first identify the factors which influence the user's perceived quality during a video streaming session and afterwards introduce objective and subjective methods for assessing and quantifying the user's QoE. Next, we start with discussing video coding in general and video coding with the compression technology H.264 in greater detail.

### 2.1. Video Coding with H.264

Video coding is the technique of efficiently compressing a sequence of pictures (or *frames*) for storage or transmission. Videos are typically recorded as a sequence of individual pictures with a rate of 24 pictures per second. Depending on the amount of change (e.g. motion, color variations) during the recorded scene, each of the captured pictures is likely to differ only slightly from the previous or subsequent pictures. Block-based compression algorithms like H.264 subdivide each picture into small blocks and try to identify how the individual blocks move between neighboring pictures. The identified dependencies within an individual picture and between subsequent pictures (intra and inter frame coding) allow for efficient coding of the video. The change (or motion) is described through *motion vectors* which require less storage than pixel data and allow to predict the subsequent picture with mostly blocks encoded in neighboring pictures. Periodical *reference* pictures are used as basis for the motion vectors. Several standards for video coding exist, in the following subsections we introduce the today's most popular video codec H.264 / Advanced Video Coding (AVC) and its scalable extension, Scalable Video Coding (SVC).

#### 2.1.1. Advanced Video Coding

H.264/MPEG-4 AVC [7] is a video coding standard developed by the Joint Video Team (JVT) [9], a cooperation between the ITU-T Video Coding Experts Group

(VCEG) [8] and the ISO/IEC JTC1 Moving Picture Experts Group (MPEG) [13], and released as final draft in May 2003. High coding and compression efficiency lead to a wide-spread adaptation of the standard in today's Internet. In the following we give a brief introduction to the prediction hierarchy employed by H.264.

H.264 defines three types of picture frames for the encoded picture stream, *I*, *B* and *P* frames. *I*, or *inter* frames are reference frames, i.e. they can be encoded independently of other pictures and do not rely on motion vectors. *P*, or *forward-predicted*, frames depend on a preceding I-frame for decoding and consist of motion vectors and, if necessary, additional encoded pixel data. The encoded pixel data is required in cases where it is not sufficient or uneconomical to use only motion vectors for the description of the picture. P-frames can also use other preceding P-frames as reference. *B* (*bidirectional prediction*) frames are similar to P-frames, but in addition to the reference options available to P-frames, they can reference to preceding and subsequent P and B frames. A series of a specific number of I, P and B-frames, with at least one reference frame (i.e. I-frame), is called Group-Of-Pictures (GOP). A GOP is self-contained and includes all data to decode the time slice the GOP represents. The number, size and structure of a GOP is not fixed and can be configured during the encoding process. For adaptive streaming of video content encoded with H.264, the GOP size dictates the smallest possible unit for segmentation.

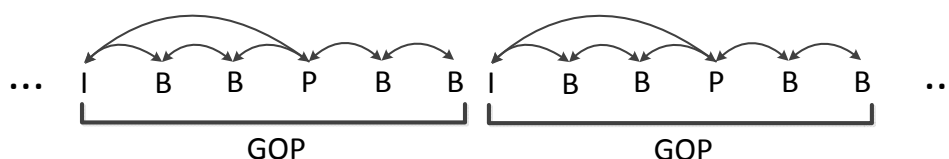


Figure 2.1.: Example AVC GOP structure

Figure 2.1 illustrates a possible simple GOP structure. The GOP is made of one I frame, one P frame referencing the I frame and four B frames referencing each other and the P and I frame. The GOP structure and size can have a significant effect on the resulting bitrate of the compressed video content [21].

### 2.1.2. Scalable Video Coding

Scalable Video Coding (SVC) was developed and specified as Annex.G of the H.264 / AVC video compression standard by the Joint Video Team (ITU-T VCEG, ISO/IEC MPEG) [56]. SVC encodes the content into a *bitstream* with multiple substreams where the different substreams can be accessed by dropping parts of the bitstream. SVC provides three scalability options. Spatial scalability allows for switching to a different resolution, temporal scalability enables the adaptation of the frame rate and quality scalability increases and decreases the fidelity of the content. The provided scalability options allow for the on-the-fly adaptation of the stream to different network conditions and device capabilities.

In SVC, valid substreams are also called *layers*. A valid substreams contains at least the *base-layer*, a AVC-compatible substream which represents the lowest

temporal, spatial and quality level of the stream and zero or more *enhancement layers*. An enhancement layer always depends on the previous enhancement layer(s) of that particular scalability dimension. Figure 2.2 illustrates the different scalability options by example in a three dimensional space using "subcubes".

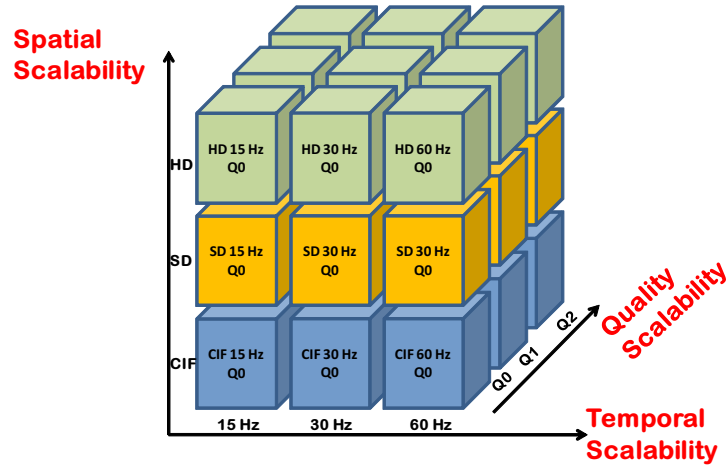


Figure 2.2.: Possible Scalability Options for SVC ([66])

The example shows a SVC bitstream with two enhancement layers for each SVC scalability dimension. The resulting bitstream contains three quality (Q0, Q1, Q2), spatial (CIF, SD, HD) and temporal (15 Hz, 30 Hz, 60 Hz) layers. The base-layer subcube is located closest to the origin of the coordinate systems and represents the content in CIF resolution, 15 frames per seconds and a quality level of 0. Increasing the resolution to HD and the frame-rate to 60 frames per seconds would require all labeled subcubes in the front, a total of 9 subcubes. Further increasing the quality by using quality level 3 instead of 0 would require all 27 subcubes.

The supported scalability options come with a cost of coding efficiency compared to single layer coding. Coding efficiency depends highly on the used encoder configuration and the type of content and can not be compared reliable in general. Studies predict a coding efficiency penalty of approximately 10% - 20 % on average for each added spatial layer [58, 63, 57] compared to single layer coding.

## 2.2. MPEG-DASH for Video Streaming

In the following, we introduce Dynamic Adaptive Streaming over HTTP (MPEG-DASH), a standard for streaming media content over the Internet using the HTTP protocol. First, we discuss progressive download for HTTP, the underlying principle of video streaming over HTTP. With the success of video portals like YouTube, progressive download over HTTP has become the most dominant technique for media content delivery over today's Internet. Next, we give a brief introduction to DASH, which extends the progressive download principle to allow for adaptation to the current network and viewing environment. Third, we compare DASH to datagram-based streaming techniques. In the last subsection, we introduce three investigated DASH adaptation algorithms taken from the literature.

### 2.2.1. HTTP Progressive Download

Progressive downloading describes the process of continuously transferring data from a server to a client, typically through the HTTP protocol. In the context of media delivery, progressive download is the approach of making the media bitstream available on a HTTP server for clients to request. The client issues a standard HTTP request and the server sends the bitstream data to the client. Next, the media player on the client side buffers a specific amount of data to compensate for short bandwidth fluctuations and subsequently begins to present the content to the consumer. However, an ordinary HTTP server is content unaware and treats the media bitstream equal to other files (e.g. text, images, compressed files). Accordingly, the media content is delivered using best-effort with respect to the available resources and the client and server are unable to adapt the transfer to the actual media bitrate. Hence, a sending rate lower than the media bitrate leads to buffer starvation (i.e. stalling), a sending rate exceeding the bitrate requires artificial traffic shaping through the TCP congestion control algorithm or large buffers. The impact of stalling and ineffective use of resources on the Quality of Experience as perceived by a human viewer and on the Quality of Service are discussed in Section 2.4. In the subsequent subsection, we describe how DASH extends the progressive downloading concept for media delivery to allow for adaptive media streaming.

### 2.2.2. MPEG - Dynamic Adaptive Streaming over HTTP

In recent years, the lack of a common standard for adaptive video streaming over HTTP lead to the development of commercial and proprietary streaming solutions like Adobe HTTP Dynamic Streaming [20], Apple HTTP Live Streaming [22] and Microsoft Smooth Streaming [45]. As a result of this development, video streaming devices have to support multiple protocols to access different streaming services and users are often limited to the streaming client supplied by the streaming provider. A common standard for HTTP video streaming would allow standard-compliant client devices to access any standard-compliant video streaming service. Therefore, MPEG-DASH is intended to provide a common standard for HTTP video streaming over the Internet. The work on Dynamic Streaming over HTTP (DASH) started in April 2009 when the Moving Picture Experts Group (MPEG [13]) announced a Call for Proposal to create a HTTP streaming standard. Three years later, in April 2012, the standard was published as ISO/IEC 23009-1 [5].

Dynamic Streaming over HTTP (DASH) defines a control protocol for content which is *a)* split into short segments, each representing a none overlapping time slice of the content; *b)* where each segment is encoded in different alternatives (e.g. different resolutions); and *c)* stored on a HTTP server. The client can choose for each time slice of the content which alternative to download and display. The DASH standard is not limited to using different resolutions for the segments, but also different audio tracks, subtitles, encoding parameters and further options are supported. Which video codec to choose and how the client should adapt the playback based on the options offered, is out of scope of the DASH standard and therefore left to content providers and client implementations to decide.

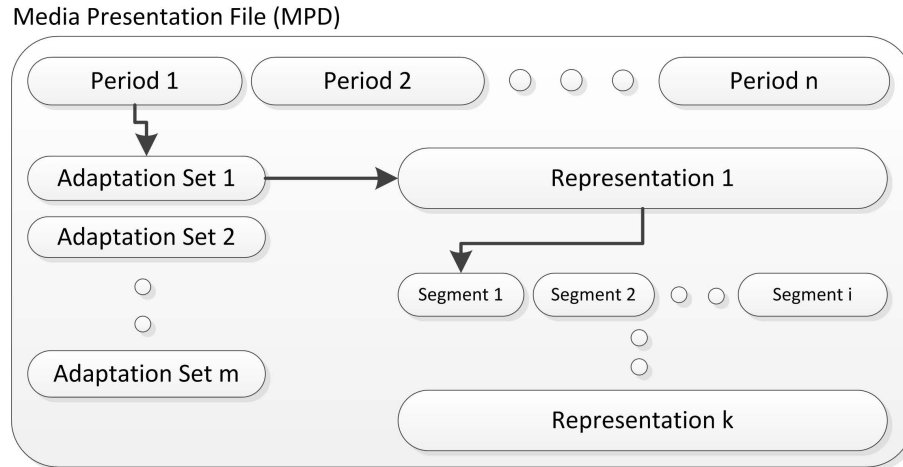


Figure 2.3.: Media Description File (MPD)

A Media Presentation Description (MPD) file is specified to describe the properties and URLs of the content and its segments. Based on the information supplied by the MPD file the client is able to adapt the playback to the current viewing environment (e.g. display size, available network bandwidth). The MPD file is structured as illustrated in Figure 2.3. On the highest level in the hierarchy, the media is segmented in *Periods*. A period represents a time period where the set of adaptation options does not change. For instance, a period could contain the main movie with several adaptation options, but a second period comprised of out-takes is only available with a reduced set of options. An *Adaptation Set* is a logical group of adaptation options. Typically, there are three adaptation sets defined for a full-length movie, one for the video, one for the audio and one for the subtitle adaptation options. An adaptation set in turn contains different representations of the specified option. For instance, in terms of a video adaptation set, representations can equal specific content resolutions. The end of the hierarchy marks the *Media Segments*, which contain the location (i.e. URL) of the described media content segments in chronological order.

### 2.2.3. Comparison to Datagram-based Streaming

In the following, we compare TCP-based video streaming (e.g. HTTP/DASH streaming) with datagram-based streaming (e.g. RTP over UDP). Relevant factors for the comparison are summarized in Table 2.1 for a typical use case. As transport protocol, datagram-based streaming services typically utilize the UDP protocol, whereas HTTP streaming is based on the stream-orientated TCP transport protocol. UDP is a connectionless and unreliable transport protocol where delivery and delivery order of the packets is not guaranteed. TCP in turn offers reliable and in-order delivery of the data. The UDP transport protocol allows RTP to support multicast traffic, where a group of nodes in a network can be reached simultaneously and more efficiently by one server. HTTP streaming in turn is limited to unicast traffic, where each client has to open a separate connection to the streaming server. However, with streaming over HTTP the session is managed by the client and does not face

Property	RTP/UDP streaming	HTTP/TCP streaming
Transport Protocol	UDP	TCP
Supported Topologies	Unicast, Multicast	Unicast
Content Source	Live, pre-encoded	Live, pre-encoded
Overhead	Low	Medium
Consequence of insufficient bandwidth	Image distortions, Stalling	Stalling
Delay	Low	Medium to High
Session management	Server, Client	Client
Firewall/NAT friendly	No	Yes
Congestion control	No	Yes

Table 2.1.: Comparison of DASH/HTTP and RTP/UDP

the scalability issues server-side session management, as generally employed by RTP streaming, implies [30, 54]. Both streaming techniques support live and pre-encoded content. Regarding protocol overhead, RTP can use the available bandwidth more efficiently than TCP due to the low overhead of the UDP protocol compared to TCP. This overhead and longer delay is due to the reliable transport ability of TCP, where guarantee of data delivery, packet reordering and congestion avoidance is built into the protocol. In scenarios with multiple clients sharing one Internet connection, the congestion avoidance feature may lead to a fair bandwidth sharing between two video streaming sessions, whereas with UDP streaming, the behavior between two clients is undefined. However, studies show, that with UDP-streaming, a fair bandwidth sharing is also possible [28]. As consequence of insufficient bandwidth, HTTP/TCP streaming suffers from buffer underruns (i.e. stallings), if the receiving bandwidth drops underneath the bitrate of the video and the playback buffer is depleted. RTP/UDP in turn exhibits image distortions due to packet loss in addition to possible stalling events. HTTP/TCP data transfers are well supported in today's Internet infrastructure and therefore HTTP/TCP streaming is possible in most environments where firewall and NAT devices are present.

#### 2.2.4. Investigated Adaptation Algorithms

The MPEG-DASH standard does not specify how a DASH client implementation should adapt the playback to the available bandwidth. As a consequence, the choice of the adaptation algorithm is left to the implementation. However, we show in this thesis that the selection of the adaptation algorithm dictates the resulting playback behavior and therefore also the Quality of Experience of the viewer. Thus, to maximize the QoE of the viewer, the choice of the adaptation algorithm is of significance. In this thesis we implement and evaluate three published adaptation algorithms and compare them to the algorithm proposed by the author of this thesis. The following sections give a brief introduction to the three algorithms taken from publications.



### 2.2.4.1. TRDA

The AVC-based adaptation algorithm proposed and evaluated in [46], henceforth referred to as TRDA (TUB Receiver Driven Adaptation, as named by the author of this thesis), uses an estimation of the current bandwidth, the current buffer level, and the average bitrate of the different representations to decide which quality level to select at a given time. The decision process is accompanied by a set of constants which have to be tuned for proper operation of the algorithm. A set of suggested default values are proposed by the authors of the publication.

$$0 < B_{min} < B_{low} < B_{high}, \quad 0 \leq B_{curr} \quad (2.1)$$

In the following we describe the behavior of the algorithm dependent on the current buffer level  $B_{curr}$  and the adjustable buffer limits  $B_{min}$ ,  $B_{low}$  and  $B_{high}$  with the constraints specified in Equation 2.1.

$$B_{curr} \in [0, B_{min}]$$

A buffer level lower than  $B_{min}$  is considered critical for the playback regardless of the currently available bandwidth. In such a case, the algorithm instantly switches to the lowest representation to avoid buffer starvation and thus playback stalling.

$$B_{curr} \in [B_{min}, B_{low}]$$

Depending on the available bandwidth, the algorithm either stays on the current quality level or decreases the quality level by one. If the currently available bandwidth is not enough for the average bitrate of the currently selected representation, the algorithm decreases the quality level. Otherwise the current quality level is kept. This is done to prevent unnecessary quality switches in cases where the buffer level is low, but is likely to increase again soon.

$$B_{curr} \in [B_{low}, B_{high}]$$

To avoid undesired quality switches, the algorithm does not switch the representation for this buffer level interval. Additionally, if the currently estimated bandwidth is not enough for the next higher representation, the algorithm adds an artificial delay to the download queue to inhibit buffer growths.

$$B_{curr} > B_{high}$$

If the currently estimated bandwidth is higher than the bitrate of the next higher representation, the algorithm switches to the next higher representation. If the bandwidth is not sufficient the algorithm introduces an artificial delay to the download queue.

In addition to the mentioned normal mode of operation, the authors introduced a more aggressive fast start-phase to speed up the adaptation process at the start of the playback. For a description of the fast start phase refer to the publication.

### 2.2.4.2. KLUDCP

In [47], the authors propose and evaluate a DASH-AVC adaptation algorithm, in the following referred to as *KLUDCP*, in a vehicular mobility scenario. From the description of the algorithm follows that the algorithm's decision depends on the average available bandwidth measured during the download of the latest segment, the average bitrate of the different quality levels and the current buffer level. The algorithm takes one configuration parameter, the desired buffer level.

Next, we give a brief description of the algorithm's method for choosing the quality level for a segment  $i$  after segment  $i - 1$  has finished downloading. The throughput is continuously monitored during the download of segment  $i - 1$  and the average throughput is used as an estimation of the currently available bandwidth for the download of segment  $i$ . Additionally, the current buffer level is used to adjust this estimation by decreasing the estimation by a constant factor if the buffer level is less than 35% and increasing it if the buffer level is higher or equal to 50% with the goal to keep the buffer at half of the maximum capacity. The resulting estimated available bandwidth is compared to the average bitrates of the available quality levels and the quality level with an average bitrate less or equal to the available bandwidth estimation is chosen for segment  $i$ .

### 2.2.4.3. Tribler

In [50] the authors propose a SVC-based adaptation strategy for distributing scalable video content in P2P systems, henceforth referred to as *Tribler*. We adopted the algorithm for DASH by defining a fixed segment downloading order and allowing only one simultaneous segment download.

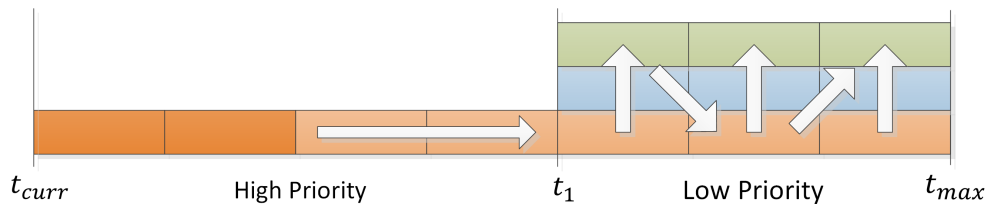


Figure 2.4.: Tribler Download Strategy

We implemented the algorithm as shown in Figure 2.4. The algorithm takes two configuration parameters.  $t_1$  is the size in seconds of the *high priority set* and  $t_{max}$  the size of the *low priority set* including  $t_1$ . The high priority set is a list of time segments starting from the current playback position for which only the base layer segments are downloaded. If all segments from the current high priority set are already downloaded and buffered, the algorithm starts to download the segments from all qualities levels between  $t_1$  and  $t_{max}$  in vertical order. During start-up, the algorithm starts the playback when all segments from the high priority set are locally available for playback. If all segments from both priority sets are already buffered, the algorithm idles until new segments are added to the low priority set (i.e. when the current playback position moves forward).

## 2.3. Scalable Video Coding with DASH

In what follows, we discuss the advantages and drawbacks of using scalable video coding for DASH as opposed to using single layer coding. Utilizing scalable video coding for encoding DASH content *a)* increases the cache-hit-ratio for HTTP caching servers and reduces the required storage space for DASH video content; and *b)* allows for greater flexibility during segment selection compared to single layer coding. Caching servers are introduced by providers to store frequently requested HTTP content closer to the user and this way prevent expensive long-haul traffic. However, caching servers have a limited capacity and can only store a subset of the requested data. Hence, caching servers have to discard uneconomically (e.g. infrequently requested) content. In terms of DASH video streaming, with single layer coding, caching servers can store the most popular quality levels of a specific video content. However, with scalable video coding, the caching server is able to take advantage of the layered coding schema where quality levels are additive to each other. Accordingly, between single layer and scalable encoded content with equivalent quality levels, scalable video coding can store a higher number of quality levels using an equal amount of storage capacity [55].

In terms of adaptation options, scalable video coding increases the flexibility for the segment selection process during the streaming session. With single layer coding, the decision which quality level to download next is effectively limited to the time slice subsequent to the current buffer position and the decision can not be changed afterwards without discarding the already downloaded segment. Whereas with scalable video coding, the algorithm can first download base layer segments and later upgrade specific time slices with additional enhancement layers.

In context of DASH, scalable video coding faces two drawbacks compared to single layer coding. *a)* A coding efficiency penalty and *b)* a higher number of segments. The penalty on coding efficiency observed for scalable video coding increases for every additional quality level. However, there are two feasible best practices to limit the coding penalty of scalable video coding for adaptive HTTP streaming. First, only make use of encoding parameters known to produce output with a high coding efficiency (e.g. dyadic resolutions). Second, instead of offering one bitstream with many scalability options, content provider can offer assorted bitstreams with a reduced set of quality levels. For example, the provider could provide device-type-specific bitstreams for each of the three categories smartphones, tables and HD-TV. Both approaches reduce the flexibility of scalable video coding, but in turn reduce the impact of the coding efficiency penalty. For an equal number of quality levels, scalable video coded content is split into a greater number of segments than with single layer coding, because each quality level requires all the lower quality levels up to the selected quality level. Accordingly, this increases the number of required HTTP requests and therefore the percentage of HTTP overhead in relation to the content size. However, in our experiments and the considered scenarios the amount of HTTP overhead was not more than 1%<sup>1</sup> for the lowest quality layer. Furthermore, the overhead may be further reduced by compression and/or use of

---

<sup>1</sup>For an average GET request overhead of 700 Bytes [6] and an average segment size of 599 Kilobytes

the new HTTP 2.0 standard [18], which is designed to reduce the HTTP overhead for multiple subsequent HTTP requests.

## 2.4. QoE of Video Streaming

The Quality of Experience (QoE) of video playback as perceived by a human viewer depends on multiple factors. The encoding of the content and type of the delivery (e.g. live or VoD [27]), the viewing environment (e.g. HD-TV at home or mobile device in public transport) and the individual user expectations [25] dictate the satisfaction of the user with the service. In the following, we first discuss how to classify video content by its temporal and spatial information and how different temporal and spatial information influences the QoE. Next, we identify three significant quality influence factors of adaptive HTTP video streaming. Namely, temporal impairments (i.e. stallings), video quality and quality flicker effects due to the adaptation process. We conclude the section by discussing established approaches for quantifying the QoE of a viewer based on the identified factors.

### 2.4.1. Temporal / Spatial Information

*Spatial Information* describes the level of detail (or fidelity) of a single image. *Temporal* information describes the degree of similarity between two subsequent images of a given image sequence. The spatial and temporal information index can be used to characterize video content. A high degree of fidelity is found in complex scenes with fine details, sharp edges and multiple objects, whereas low spatial information indicates large areas of similar colors, few objects and smooth transitions between surfaces. A high amount of temporal information in an image sequence indicates fast and frequent motions and scene changes.

ITU-T Recommendation P.910 *Subjective video quality assessment methods for multimedia applications* [40] defines the temporal and spatial information index as follows. The luminance plane of an image  $n$  in a sequence of images is denoted as  $F_n$  and the *Sobel* [17] operator is an edge-detection algorithm.

$$\textit{Spatial Information (SI)} = \max\{\textit{std}[\textit{Sobel}(F_n)]\}$$

$$M_n(i, j) = F_n(i, j) - F_{n-1}(i, j)$$

$$\textit{Temporal Information (TI)} = \max\{\textit{std}[M_n(i, j)]\}$$

The content type plays an important role in the human perception of video playback and therefore can not be neglected in the design of QoE evaluation studies. However, the implications of the different content types on the perceived QoE are not fully understood. Nevertheless, some effects of different temporal and spatial properties were identified by user studies to have an impact on specific quality influence factors. For example, the fidelity of a video sequence correlates to the perception of video quality where a high level of detail requires high level of video quality to

please a human viewer [31]. In the following subsections, we introduce three quality influence factors of adaptive HTTP video streaming and highlight known effects of different content types on the described factors.

## 2.4.2. Influence Factors of SVC-based Video Streaming

The quality of video streaming content as perceived by the user depends on many factors. In the following, we discuss the influence of the temporal and spatial information index, video quality as perceived by a viewer, playback interruptions and time-varying quality of the content on the user's Quality of Experience.

### 2.4.2.1. Temporal Impairments

Temporal Impairments (or *stalling*) effects are distinguished by their time of occurrence. Stalling at the beginning of the playback is called *initial delay*, whereas stalling during the video playback session is called *playback interruption* or just *stalling*. Initial delay is due to pre-buffering part of the content to compensate for bandwidth fluctuations during the playback. In the context of HTTP/TCP-based video streaming, playback interruptions occur in situations where the available bandwidth is not sufficient for the current content bit rate. Playback interruptions have a significant effect on the user's perceived QoE, whereas initial delay is more likely to be accepted by the user [36, 34, 37]. Stalling can be quantified by the lengths of the stalling events and their frequency during the playback session.

### 2.4.2.2. Video Quality

Scalable video coding allows for on-the-fly adaptation of image resolution, frame rate and image quality of a video sequence to the current network and viewing environment. Next, we discuss the influence of the three scalability dimensions on the video quality (i.e. Quality of Experience (QoE)) as perceived by a human viewer. It is obvious that the user is less likely to accept the provided video quality if the resolution, image quality and frame rate is low and he is more likely to accept it if all three dimensions offer a high quality. The influence of different quality levels of the different dimensions is more complex and less obvious.

Figure 2.5 gives a schematic of this issue. There exists a 3-tuple which represents the minimum quality for each dimension the user is likely to accept. Decreasing the quality along any dimension results in a QoE the user does not accept. However, a very high quality along one dimension may compensate for a very low quality on one of the other dimensions. The implications of differing quality levels for the different dimensions on the perceived QoE are not fully understood. In the following, we highlight relevant results from this research area. [29] shows, that QoE decreases non-monotonically with the video bitrate (i.e. image fidelity or resolution) and the preference of which scalability dimension to choose for the adaptation is content-dependent. The effects of adapting image quality and frame rate for different content types are discussed in [44]. The study shows that sport-coverage with a high amount of motion does not necessary require a high frame rate, but in turn can require a

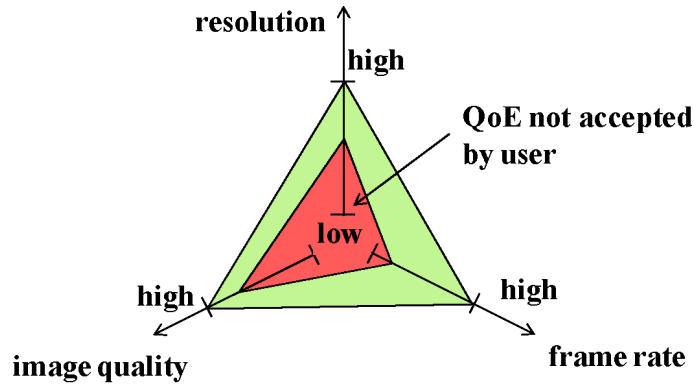


Figure 2.5.: SVC scalability options and acceptable QoE [66]

high image quality, especially on small screens. [43] performs a comprehensive user study to evaluate scalable video coding with regard to five dimensions, namely codec, content, spatial resolution, temporal resolution and image quality.

#### 2.4.2.3. Flicker Effects

Adaptive streaming can adapt the image quality and the frames per second of the content to current network conditions. The subjective impression of the varying content quality (i.e. flicker effects) is not taken into account by popular metrics like PSNR and MSE. Intuitively, the frequency and amplitude of the quality adaptations are influencing the perceived QoE, where the amplitude has a greater influence than the frequency of the switches and both should be kept low [49, 48]. In addition to the frequency and amplitude, different quality switching patterns were identified to impact the QoE differently [65]. It has also been noted, that the content plays an important role in the perception of quality changes [53]. Frequent scene changes can mask quality changes ([33]), whereas slow pan shots expose the quality changes.

### 2.4.3. QoE Assessment

There exist multiple methods for assessing the Quality of Experience of a video sequence as perceived by a user. Subjective assessments with a group of human test subjects give the best results, but are expensive and costly in terms of time. Objective algorithms try to estimate the QoE of an impaired video. The accuracy of the results compared to the subjective assessments and the computational complexity is highly dependent on the choice of the algorithm and content. The following sections take a closer look on current objective and subjective assessment methods.

#### 2.4.3.1. Video Quality

There exist several metrics for quantifying the video quality as perceived by a human viewer. Classical objective metrics like Peak Signal-To-Noise Ratio (PSNR) [14] aim to quantify degradations due to the encoding process or distortions by comparing the

individual images the video is composed of with unaltered reference pictures pixel by pixel. Evaluations show that the results of the classical error-based objective metrics perform poorly in describing the human perception of video quality [64, 60, 38]. More advanced quality metrics, like the Structural Similarity (SSIM) [60] index, focus on the structural similarity between the distorted and the reference pictures and show better results in reflecting the human perception [61]. So far motion between subsequent pictures was not taken into account. Video Quality Model (VQM) [41] is a state-of-the-art quality metric offering temporal, spatial and SNR scalability support. Temporal similarities are taken into account by processing motion vectors between subsequent images. The objective methods described here belong to the category of *full-reference* (FR) metrics. FR metrics compare an unimpaired reference sequence with the distorted sequence to measure the quality degradation and are generally used in laboratory studies where both, the reference and distorted sequence are easily available.

### PSNR

Peak Signal to Noise Ratio (PSNR) describes the impairment of a signal by calculating the ratio between the maximal signal output and the recorded noise responsible for the impairment. In the case of image compression, PSNR compares the luminance of a reference picture with the distorted image pixel-by-pixel. The PSNR of a sequence of images is the mean of all PSNR values of all images in the sequence. Because of its ease of use and low computational complexity, PSNR is the most commonly used objective image and video quality metric. However, evaluation show a low correlation to the human perception [64, 60, 38].

### SSIM

The Structural Similarity Index Metric (SSIM) is designed to reflect the properties of the human visual system by taking the structure of the image in account. Like PSNR, the SSIM metric is calculated for each picture of a sequence individually. The index is derived from the combination of luminance, contrast and structure measurements. Accordingly, both images are compared by these three aspects and the results are combined to one index between one (being the best possibly similarity) and zero (no similarity). Relatively low computational complexity allows for real-time implementations [26]. Two extensions exist to the presented (simple) SSIM, *Speed-SSIM* and *Multiscale SSIM* (MS-SSIM). Speed SSIM extends the SSIM concept with statistical models of human visual motion perception. MS-SSIM proposed in [62] utilizes multiple weighted scales of the image to take different viewing conditions in account.

### VQM

The Video Quality Model (VQM) [41] metric offers full scalability support and high correlation to the human perception with the cost of high computational complexity. In contrast to SSIM and PSNR, VQM is designed to include the spatial, temporal and signal-to-noise ratio (SNR) scalability, which is found in the Scalable Video Coding annex of AVC. The algorithm implements these aspects by considering the frame rate, SNR and motion vectors of the image

sequence. The result is a value between zero (no perceptible impairments) and one (maximal perceptible impairments).

In [59], comprehensive user studies were conducted to evaluate the performance of the presented quality metrics in regard of their accurateness to the human perception of video quality. Next, we highlight relevant results of this study. First of all, existing assumptions of the bad performance of PSNR are validated. Out of the investigated quality metrics, the outputs of PSNR show by far the lowest correlation to human perception. The study also points out the advantage of algorithms which take motion between pictures of the sequence in account. For example, Speed SSIM can considerably improve the results of SSIM through statistical motion models. VQM shows the highest correlation to human perception out of the three given metrics. However, in this thesis we utilize (simple) SSIM for quantifying different quality levels of video sequences because of its low computational complexity. Next, we discuss subjective methods for quantifying the QoE of a video sequence.

#### 2.4.3.2. Subjective Methods

Subjective quality assessment methods generally consist of viewing sessions where a group of human evaluators watch and rate video sequences based on their individual subjective judgment. Different recommendations exist how to prepare the viewing environment, how to structure the test sessions and how to evaluate the collected data. ITU-T Recommendations P.910 *Subjective video quality assessment methods for multimedia applications* [40] and BT.500-11 *Methodology for the subjective assessment of the quality of television pictures* [24] are two of the most complied with specifications for subjective quality assessments. In the following, we introduce the Absolute Category Rating (ACR), also called Single Stimulus Method, which we use for the design of our user studies in this thesis. Afterwards, we compare the ACR method to other quality assessment methods.

Absolute Category Rating describes a mode of operation for user studies where test participants are presented with individual and independent stimuli (i.e. video sequences) in random order. Each stimulus is followed by a period of voting where the subjects judge the perceived quality during the stimulus on a predefined scale. Figure 2.6 gives an example for an ACR test session. The excerpt shows a session with four test sequences (b, c, d, e) in random order. Each sequence has a length of 15 seconds and is followed by voting period. Note that the ITU Rec. P.910 recommends to limit the voting time period to ten seconds. However, limited voting periods are not supported by the user study framework used in our evaluations and therefore we omitted the time limit in the crowdsourcing campaigns.

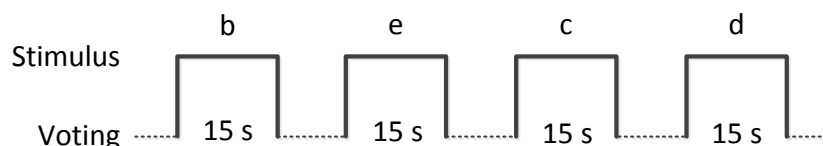


Figure 2.6.: Example excerpt from an ACR test session



The ITU recommendation introduces different predefined quality rating scales. In our evaluations, we utilize the five-point rating scale presented in Table 2.2 ranging from 5 (*excellent*, respectively *imperceptible* for impairments) to 1 (*bad* and *Very annoying*). This scale is equal to the Mean-Opinion-Score (MOS) [39], a popular metric to describe the satisfaction of a group of human evaluators with a service.

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

Table 2.2.: Five-point rating scale

Next, we give a brief introduction to other popular assessment methods suggested in the P.910 and BT.500 recommendations. With *Degradation Category Rating* (DCR) suggested in Rec. P.910, a reference video sequence is presented in conjunction with impaired sequences. Viewers rate the amount of impairment. *Pair Comparison* (PC), also presented in Rec. P.910, is a method where pairs of video sequences with the same content, but different impairments, are presented and viewers rate which one they prefer. With *Single stimulus continuous quality evaluation* (SSCQE) described in BT.500, viewers are presented with a continuous playback of a video sequence with time-varying impairments. With the means of a slider, the test subjects are asked to continuously rate their viewing experience. No reference sequence is provided. Also suggested in BT.500, *Double-stimulus continuous quality-scale* (DSCQS) describes a method where short sequences consisting of reference and impaired content sequences are presented in random order. The human evaluators are asked to rate each sequence individually.

#### 2.4.3.3. Crowdsourcing for QoE Assessment

Evaluating the Quality of Experience of video content is costly in terms of money, time and facilities. Crowdsourcing can help to reduce the cost compared to laboratory studies. In essence, crowdsourcing for QoE uses the Internet to utilize anonymous test subjects for online-based studies. The studies can be completed at home, are usually browser-based and do not require any special equipment or competencies. Furthermore, online platforms like Amazon’s Mechanical Turk [1] and Microworkers.com [12] help the researcher to make QoE studies available to a larger crowd. To create a study and make it publicly available the researcher has to specify the URL of the study, add a brief description, specify the time requirement and the monetary compensation. Interested workers sign up to receive a list of currently available and suitable studies, from which they can choose their next task. In contrast to studies in controlled environments with selected participants, crowdsourcing can not guarantee the reliability of the anonymous test subjects and is more prone to cheaters. Special measures have to be taken to filter out invalid results [35].

## 3. Subjective Studies

In the following, we discuss the subjective studies conducted in the course of this thesis. We first interrelate the studies to the objective performance evaluation. In the objective performance evaluation, we introduce metrics to assess and compare the different adaptation algorithms from a user-centric and resource-centric point of view. Most of the user-centric metrics like playback quality, initial delay and stallings are related to common non-adaptive media-streaming techniques and therefore the influence of these metrics on the user's perceived QoE is well understood. However, little information exists about the correlation between the quality switches found in adaptive video streaming scenarios and their influence on the actual perceived QoE of the user. The conducted subjective studies are designed to gain a deeper understanding of the influence of the quality switches on the user's QoE.

Figure 3.1 gives an overview of the conducted subjective studies and also illustrates the structure of this chapter. We first formulate the problem at hand through a set of research and control questions, each question dealing with a specific aspect of the quality switches (e.g. influence of switching amplitude) and accompanied by a hypothesis (e.g. the amplitude greatly influences the QoE). Based on the formulated questions, we implemented a pilot study in a laboratory as preparation for the crowd-sourcing campaigns. In the pilot study, we asked a group of experts to assess 20 test scenes with different number of quality switches, switching amplitudes and varying amount of motion and image detail. The findings of the pilot study helped to select test scenes and reasonable starting values for parameters (e.g. the amplitude for the switches). Based on the formulated questions and the pilot study, we designed and implemented five crowd-sourcing campaigns. The design of each campaign was guided by one or more of the formulated research questions. Through a crowd-sourcing provider we made the campaigns available to a large international crowd and each campaign was completed by about 100 participants. After filtering unreliable submissions [35], we identified the relevant QoE influence factors through statistical analysis by utilizing, among others, main effect plots.

The contribution of the subjective studies to the understanding of the influence of quality switches on the perceived QoE is threefold. First, we present findings concerning the perception of quality switches. In particular, we take a look at the question if the participants were able to accurately guess the number of quality switches in a test scene. Next, we highlight the observed impact of quality switches on the user's QoE in terms of a Mean Opinion Score (MOS) in the range of 0 to 1000 (continuous quality scale). Third, we discuss the acceptance of quality switches in the evaluated test scenes. Afterwards, we summarize the identified Quality of Experience influence factors and discuss how the findings help to assess the objective performance results and how to design a user-centric DASH adaption algorithm. In the subsequent section we introduce the research and control questions.

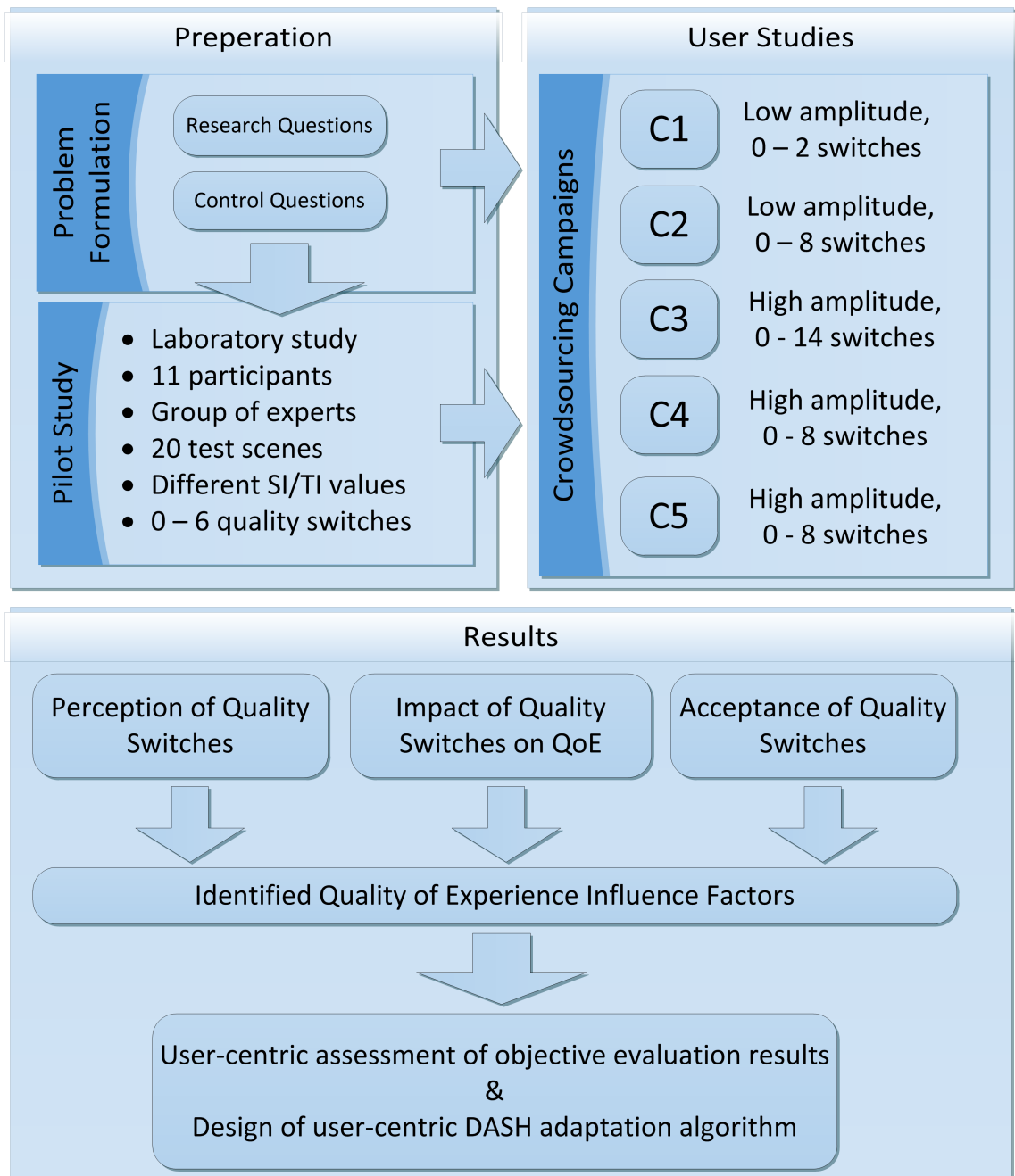


Figure 3.1.: Subjective studies overview with different research questions examined through different crowdsourcing campaigns (C1 - C5).

### 3.1. Problem Formulation

As we will see later in Chapter 4, the switching frequency, the amplitude of the switches and average playback quality is highly dependent on the choice of the adaptation algorithm. In order to evaluate the algorithms from the point of view of the user, it is therefore necessary to gain insight of the aforementioned metrics regarding their effect on the user’s perceived Quality of Experience. The effect of

the presented playback quality is well understood, but there are open questions regarding the influence of quality switches. In particular, for a specific switching amplitude, how many quality switches are tolerated by a user during a specific time period of the playback session before he judges the video playback quality as unacceptable. Furthermore, if the algorithm has to choose between presenting a continuous low playback quality or fluctuating between the low playback quality and a higher quality, which alternative is more likely to be accepted by the user. To evaluate the problem regarding these questions, we ask the user about two different aspects, overall quality and acceptance. In particular we ask *How would you rate the overall viewing experience?* and *Would you use a website offering this service quality?* The acceptance rate gives the percentage of user's who would accept the provided quality. The overall viewing experience rating is explained in Subsection 3.3.1, where we describe the web-based user interface, in greater detail.

In the following, we first discuss the research questions the user studies are based on. Afterwards, we discuss control questions which were, in addition to the research question, also considered for the design of the studies.

#### 3.1.1. Research Questions

The user studies are designed to answer the following research questions. First, we want to know if the content influences the perception of the quality switches. We hypothesize that the perception, and therefore also the QoE of the user for scenes with quality switches, depends on the temporal and spatial information of the content. We address this question by the pilot study and the campaign C1, where we use scenes with different SI/TI values, but the same number of quality switches. Next, we ask if the amplitude influences the QoE of the user. We surmise, that quality switches with a higher amplitude have a greater impact on the user's QoE than switches with a low amplitude. To test this hypothesis, we include quality switches with a low amplitude into the campaigns C1 and C2, whereas the switches in the campaigns C3 - C5 have a high amplitude. The last research question asks how many quality switches per 30 seconds are acceptable for a user. Our hypothesis is, that the user does not tolerate more than four quality switches per 30 seconds. We consider this question in all five user studies and in the pilot study by using switching patterns with multiple different number of quality switches.

#### 3.1.2. Control Questions

The following control questions are considered by the design of the crowdsourcing user study. First, are the study participants able to estimate how many quality switches occurred during playback of a test sequence? We hypothesize that the users are able to tell whether there have been any quality switches in the sequence but can not accurately estimate the number of switches. To test the hypothesis, we ask the user after each test sequence to guess how many quality switches have occurred during playback. Second, we want to know, if there are two test sequence which differ only in the average playback quality, whether the one with the higher average quality gets rated differently. We assume, that the sequence with a higher

Question	C1	C2	C3	C4	C5	Pilot
Does the content influence the perception of the quality switches?	X					X
Does the amplitude of the quality switches influences the QoE of the user?	X	X	X	X	X	X
How many quality switches per 30 seconds are acceptable for a user?	X	X	X	X	X	X
Are the study participants able to estimate how many quality switches occurred during playback of a test sequence?	X	X	X	X		
If there are two test sequence which differ only in the average playback quality, does the one with the higher average quality gets rated differently?	X	X	X	X	X	

Table 3.1.: Research questions implemented per campaign

average playback quality is rated better. To test this hypothesis, we vary the distribution of the quality switches between the test sequences to create sequences with differing percentage of time spent on the highest quality level.

Table 3.1 summarizes the research and control questions and gives an overview of the questions considered for each of the crowdsourcing campaigns and the pilot study. Scenes with different SI/TI values were only used in C1 and the pilot study. In the pilot study, the user were not asked to guess the number of quality switches and no variations in average playback quality were assessed. In C5, the users were not asked to guess the number of quality switches.

## 3.2. Pilot Study in a Laboratory

We conducted a user study in a laboratory at the Alpen-Adria University in Klagenfurt, Austria to answer the following two questions as preparation for the crowdsourcing campaigns. First, is the temporal and spatial information index sufficient to characterize video sequences regarding their influence on the perception of quality switches? In particular, does a low amount of motion and image fidelity in a scene increase the assumed negative effect of quality switches on the QoE compared to a scene with a high amount of motion and image fidelity? And second, does the amplitude of the quality switches influence the perception of the switches? Specifically, are the amplitudes provided by our test content distinguishable by a human viewer in a typical viewing environment? And, for that matter, are quality switches with these amplitudes perceived by the test participants when watching video sequences with different SITI values? In the following we first explain how we designed the pilot study and afterwards present the results.

Segment	Time Period (Min.)	Switches	Amplitude	Special Property
1	00:00 - 00:30	0	0	no switches
2	00:30 - 01:00	4	1	
3	01:00 - 01:30	4	2	low SITI
4	01:30 - 02:00	2	2	low SITI
5	02:00 - 02:30	6	2	low SITI
6	02:30 - 03:00	1	1	
7	03:00 - 03:30	2	2	
8	03:30 - 04:00	1	2	
9	04:00 - 04:30	1	2	
10	04:30 - 05:00	0	0	no switches
11	05:00 - 05:30	5	2	
12	05:30 - 06:00	2	1	
13	06:00 - 07:30	5	2	
14	07:30 - 08:00	3	1	
15	08:00 - 08:30	4	2	
16	08:30 - 09:00	4	2	high SITI
17	09:00 - 09:30	2	2	high SITI
18	10:30 - 10:00	5	1	
19	10:00 - 10:30	3	2	
20	10:30 - 11:00	6	2	high SITI

Table 3.2.: Pilot Study: Segmented Tears Of Steel Movie

### 3.2.1. Methodology and Lab Setup

Three different test dimensions are included in the study. The frequency of quality switches, the amplitude of the quality switches and the temporal and spatial information index value. The pilot study is designed to mimic a realistic Video-on-demand viewing session. Accordingly, we do not use a small set of repeating test-scenes with different properties, but instead show the test participants a complete short-movie. To do so and still be able to test different properties, we segment the short-movie into non-overlapping time intervals of fixed lengths and assign each time interval an individual switching frequency and switching amplitude. We also identify segments with low and high spatial and temporal information (referred to as low/high SITI) and assign a specific set of properties to both low and high SITI segments.

As test content, we use the short-movie Tears Of Steel with different spatial resolutions and a segment length of 30 seconds, 20 segments in total. The properties assigned to the individual segments are presented in Table 3.2. The number of quality switches per segment ranges from zero to six. The amplitude is given in spatial resolution changes, with one referring to a change from 1280x720 (QL2) to 640x360 (QL1) and two referring to a change from 1280x720 to 320x180 (QL0). The Segment 1 and Segment 10 were presented without quality switches and in the highest quality. Compared to the other segments, Segment 3, 4 and 5 exhibit a low SITI value and the segments 16, 17 and 20 a high SITI value.

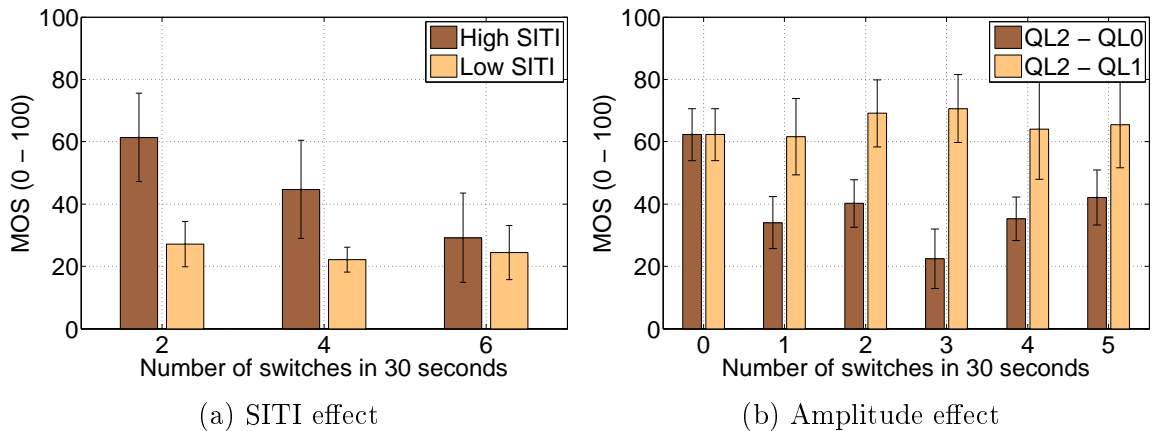


Figure 3.2.: Perception of quality switches with different SITI values and amplitudes

In the following, we briefly describe the procedure of a test session, the test environment and the demographic. First, the user is presented with an introduction explaining the test procedure and within the introduction asked to complete a short questionnaire composed of demographic questions. After the subject submitted the questionnaire, Segment 1 is presented. Next the screen displays a 0 to 100 (0 labeled as *Very low*, 100 labeled as *Very high*) rating scale slider with a default slider position of 50. The subject is given 5 seconds to change the position of the slider position if he desires. The five seconds are indicated by a visible counter. After five seconds, the rating scale disappears, the current slider position is saved as the user's rating for the presented segment and the next segment is displayed to the subject. After the rating period of Segment 20, the user study session is completed. The pilot study was conducted in a special laboratory at the University of Klagenfurt. The room was soundproofed and darkened to avoid any external influences. A standard 24 inch display was placed on an office desk and the subject seated in front of the desk. A computer mouse and keyboard was placed in front of the display to allow the subject to complete the questionnaire and to control the rating slider. Overall eleven people participated in the pilot study. All participants were recruited from the Computer And Mathematical Science department of the University of Klagenfurt, Austria and were of Austrian nationality. The average age of the participants was 29 and all were at the time of the study members of the postgraduate program in computer and mathematical science at the University of Klagenfurt. All users were male. In the following subsection, we present the results from the pilot study.

### 3.2.2. Results of the Pilot Study

The results of the segments with steady quality (i.e. Segment 1 and 10) show a reference Mean-Opinion-Score (MOS) value of 60 with a 95% confidence interval of [51.6, 68.4] for the high quality level. Next, we highlight the results of the pilot study based on the questions specified. At the end of the subsection we present the contribution of the study to the design of the crowd-sourcing studies.

Figure 3.2a addresses the question of how the temporal and spatial properties of the content influence the perceived Quality of Experience of the user. The axis on

the bottom shows the number of high amplitude switches during the playback of a segment (i.e. 30 seconds). The axis on the left shows the MOS on a scale of 0 to 100 with a 95% confidence interval. We see, that for content with low spatial and temporal information, i.e. for content with a low amount of motion and details, a number of high amplitude switches greater or equal than one negatively impact the MOS, independently of the number of switches. Accordingly, even two quality switches are perceived as low quality and the rating results for four or six quality switches do not significantly differ from the two quality switches rating. In contrast, scenes with a high amount of motion and image details reduce the negative effect of the quality switches on the QoE. Two quality switches in the scene with a high SITI value do not significantly decrease the MOS compared to the reference MOS. Four quality switches show a high uncertainty for the absolute rating, but relative to the four switches in the low SITI scene, the MOS is still higher. Six quality switches in the high and low SITI scenes are rated with an equally low MOS score. Figure 3.2b addresses the effect of the amplitude of the quality switches on the MOS. The axes are the same as for Figure 3.2a. The figure shows that one to five low amplitude switches (QL2 to QL1) do not significantly decrease the MOS. Whereas even one high amplitude switch (QL2 to QL0) decreases the MOS to an unacceptable level. Two or more high amplitude switches do not decrease the MOS further.

The contribution of the pilot study to the design of the subjective crowdsourcing studies is twofold. First, for the selection of the test scenes for the crowdsourcing studies we focus on scenes with a low SITI value. High SITI scenes exhibit a masking effect on quality switches and thus are unfit for the evaluation of the negative effect of quality switches on the QoE. Second, the pilot study shows that low amplitude switches, as specified in regard to the quality levels of our test content, are not perceived by a user for the tested number of quality switches. We therefore use high amplitude switches for the crowd-sourcing studies. In the crowdsourcing campaign Q1 we confirmed the difference between low and high amplitude switches as well as the influence of different SITI values. In the subsequent section we introduce the methodology of the crowdsourcing-based subjective studies.

### 3.3. Methodology for Crowdsourcing QoE Assessments

In the following, we discuss the methodology of the conducted crowdsourcing user studies. First, we give a general overview of the user studies and the utilized crowdsourcing approach. Next, we describe the web-based user interface used in the studies to present the test sequences, to ask the demographic questions and to gather the results. Afterwards, we introduce the test scenes. Two test scenes, one with a high amount of motion and one with a low amount of motion, are presented. Furthermore, three quality levels are specified to allow for different switching amplitudes. Next, we describe the process of filtering unreliable test subjects. At the end of this section we introduce the design of each of the five crowdsourcing campaigns conducted in the course of this thesis related to the research questions in Table 3.1.

For the user studies, we utilized crowdsourcing to reach a large and international



test crowd. Crowdsourcing is the idea of distributing short tasks (e.g. surveys, quality assessment of different compression codecs) to an anonymous online crowd for processing. A popular method to do so is through a crowdsourcing provider which supports the task creator in reaching a larger group of potential employees. For our subjective studies we used the crowdsourcing platform microworkers.com [12] for task distribution and handling of the monetary compensation of the participants. To create a task and make it available to the potential employees, three steps are sufficient. First, make the task available online and accessible by a URL. Second, specify a monetary compensation for a completed task and third, specify the maximum number of people allowed to process the task. See A.6 for details on the microworkers task description. To make the task available online, we implemented a browser-based questionnaire to present the test sequences and collect demographic information and rating results. In the following, we give a detailed description of the web-based questionnaire.

### 3.3.1. Web-based User Interface

The web-based interface of the studies was implemented using the QualityCrowd2 framework proposed by Keimal at al. [42]. The framework is designed for crowdsourcing based quality assessments through common web servers, respectively on the client-side, common web browsers. It provides a text-based scripting language to design the desired user study, incorporates anti-cheating measures and handles the generation of the payment token for finished test subjects. Next, we describe the procedure of the user study from the point of view of the test subjects.

After the test subject accepted our quality assessment task on the website of the crowd-sourcing provider, the user is presented with a custom URL which directs him to our assessment website. On the website the participant is first directed to an introduction explaining the test procedure. For example, the introduction explains how to start the playback of the test sequence and how to use the rating slider. See A.7 for details of the introduction. Next, the user is presented with a series of demographic questions about his age, education, occupation and country of residence. A detailed listing of the questions is available in A.1. Afterwards, the download of the first test sequence is started. When the download has finished, the user is able to start the playback by the push of a button. A screenshot of the video playback and rating page is available in the appendix of this thesis (A.4). After the playback of the whole test sequence, the subject uses the slider to rate the overall viewing experience. The slider is continuous, but visually split in five equally sized segments, *Excellent*, *Good*, *Fair*, *Poor* and *Bad*. After positioning the overall viewing experience slider, the subject is asked if he/she noticed any change in quality during playback and if yes, if he/she felt annoyed by the switches. The level of annoyance is, as with the viewing experience, rated by the means of a slider visually segmented in five segments. The segments are labeled *Imperceptible (did not notice any)*, *Perceptible but not annoying (did notice, but did not care)*, *Slightly annoying*, *Annoying* and *Very annoying*. Afterwards, the subject is asked to guess how many quality switches he/she noticed during the playback of the test sequence in a range of 0 to 14. The questions to this test sequence is concluded by the

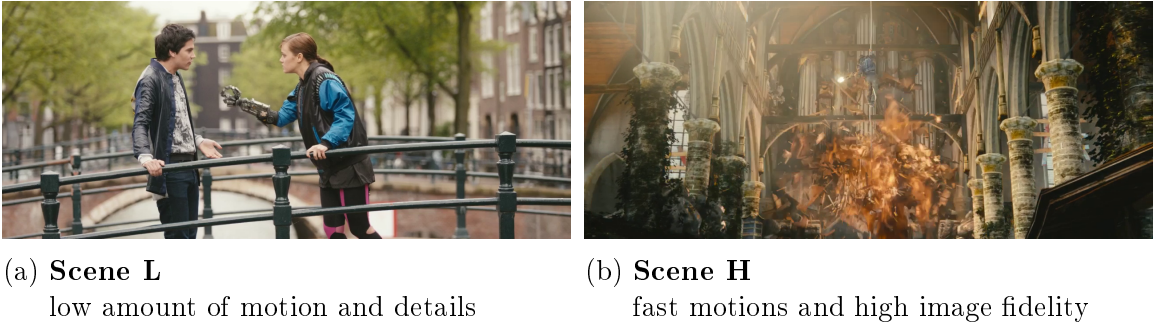


Figure 3.3.: Low and high SITI scenes used in the user studies

question, if the subject would accept a website offering this service quality. After all test sequences of the campaign are presented, the payment token is displayed to the participant, which the user can enter on the website of the crowd-sourcing provider to receive his monetary compensation of \$0.30.

### 3.3.2. Test Scenes and Quality Levels

Now, we introduce the two test scenes and the three quality levels used in the subjective studies. We selected two scenes from the short-movie *Tears Of Steel* for the user study. Scene L, depicted in Figure 3.3a, is characterized by a low amount of motion with an average SI of 8.50 and average TI of 5.38. The scene shows a couple arguing on small bridge surrounded by houses and trees. The background is blurred. The sequence is exactly 360 frames long, i.e. 15 seconds for a playback with 24 frames per second, and the start of the scene corresponds to the timestamp 00:00:25 of the full short-movie. Scene H, introduced by Figure 3.3b, exhibits fast motions and multiple scene changes with an average SI of 5.41 and average TI of 24.48. The sequence has the same length as Scene L and shows a soldier fighting against fast moving robots. As with Scene L, Scene H is taken from the short-movie, starting from the timestamp 00:08:08. We chose a length of 15 seconds for each scene based on the findings in [52], which indicate that the human memory effect for quality assessments is limited to roughly 15 seconds.

Figure 3.4 introduces the three quality levels used in the subjective crowdsourcing studies. The picture quality is here quantified by the SSIM metric. L2 is used as reference sequence and is derived from the original sequence by downscaling the sequence from a resolution of 1280x534 to 640x360. Black bars are added at the top and bottom to allow for a 4:3 aspect ratio. L0 and L1 are also based on down-scaled versions of the original sequence. L1 was created by downscaling the original sequence to 320x180 and L0 by downscaling to 160x90.

### 3.3.3. Filtering Unreliable Test Subjects

In the following section, we discuss the measures taken to identify and filter out unreliable participants during the crowd-sourcing campaigns. The measures taken aim to inhibit the two common causes of unreliable results. First, the user does not understand the questions and second, the user tries to cheat to receive the



Figure 3.4.: Different quality levels used in QoE evaluation

monetary compensation without earnestly participating in the study. The former we counteract by providing a detailed and pictorial introduction to each study (see A.7). Additionally, we use simplified English for the questions and the labeling of the rating scales. Multiple measures are taken to prevent cheating during the crowd-sourcing campaigns. First, we only provide the code required to receive the monetary compensation after the questionnaire has been completed and all questions have been answered. Second, the utilized user study framework prevents the user from skipping or fast-forwarding the presented video clips. Additionally, each rating scale has to be clicked at least once to be allowed to continue to the subsequent clip. Next, we implement easy content questions to identify unobservant test subjects. In Subsection 3.3.2, we introduce the two scenes used during the QoE evaluation in Figure 3.3. Scene L shows a human couple arguing on a small bridge. We ask the participant *Where did the protagonists stand on?* and let him choose between *A Building*, *A large field*, *A small bridge* and *Riding on an elephant*. For scene H, where a soldier is fighting large robots, we ask *The protagonist was fighting against ...* and offer *Elephants*, *Humans*, *Ducks*, *Robots* as possible answers.

To conclude this section, we implemented reasonable measures to prevent abuse of the user study for selfish monetary goals of individual users and designed the study easily accessible to reduce misunderstandings concerning the questions and the rating scales. However, it is not possible to give a success rate of each action taken due to the lack of information about the anonymous remote user. For the content questions we observed that about 11% of the participants were giving wrong answers and were therefore excluded from the evaluation. Figure 3.5 gives a detailed overview of the excluded users for each campaign. C1 contained two content questions, both had to be answered correctly. For the campaigns C1 - C5, 6, 16, 14, 8 and 4 users were excluded based on their answers given for the content question, respectively.

### 3.3.4. Crowdsourcing Campaigns

In the following, we discuss the design of the five user study campaigns conducted during the course of this thesis. Table A.5 gives an overview of the five campaigns. A more detailed overview is available in the appendix of this thesis (Appendix A.3). The individual campaigns are numbered consecutive, prefixed by the letter C. The column labeled *Switches* shows the number of switches in order as presented to

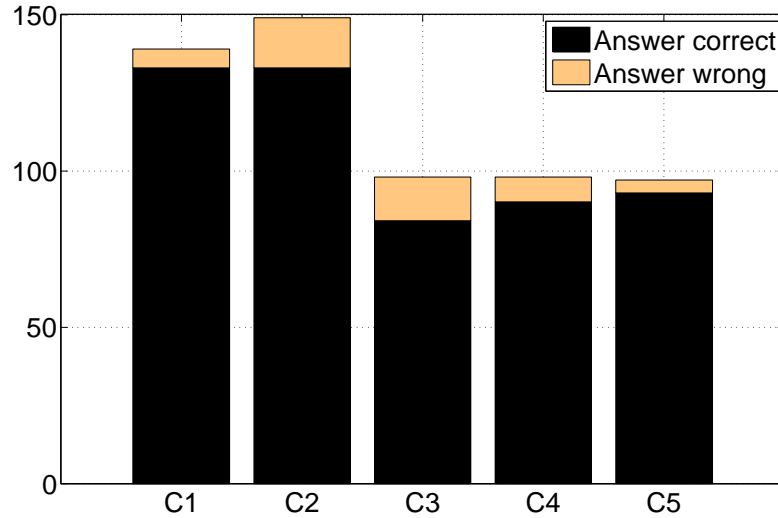


Figure 3.5.: Number of users filtered by content question

Campaign	Switches (in order of presentation)	Amplitude	Begin	Scene(s)
C1 (L)	0 (L2), 1 (L1-L2), 0 (L1), 1 (L2-L1)	L2 - L1	-	L
C1 (H)	0 (L2), 1 (L1-L2), 0 (L1), 1 (L2-L1)	L2 - L1	-	H
C2	2, 1, 8, 0 (L2), 3, 6, 4	L2 - L1	L2	L
C3	2, 1, 14, 0 (L2), 0 (L0), 3, 8, 4	L2 - L0	L2	L
C4	2, 0 (L0), 1, 7, 0 (L2), 8, 3, 5, 4	L0 - L2	L0	L
C5	2, 8, 1, 7, 0 (L2), 3, 0 (L0), 5, 4	L2 - L0	L2	L

Table 3.3.: Crowd-sourcing campaigns

the study participants. Campaign C1 is conducted for both test sequences. For the other campaigns, we only present test sequence L. The campaigns C1 and C2 use low amplitude switches (L2 - L1), whereas the campaigns C3 to C5 use high amplitude quality switches (L2 - L0). For sequences with zero quality switches, we give the used quality level in brackets. It has to be noted, that for the campaigns C1 to C4 the quality switches are distributed uniform over the 15 seconds. From this it follows that for an odd number of quality switches, the cumulative time spend on the individual quality levels differs (see A.3). Campaign C5 adjusts the distribution of the quality switches to achieve an equal amount of time for both quality levels.

### 3.4. Results of the statistical analysis of the subjective studies

Now, we present and discuss the results from the conducted user studies. First, we highlight the specific user studies used to gain the subsequent results and give the relevant properties and differences of each study. Second, we present the observed demographic of the study participants. Third, we highlight the findings regarding the perception of quality switches. Along these lines we answer the question if the

participants were able to accurately guess the number of quality switches in a test sequence. Afterwards, we show the impact of quality switches and of the time spent on the highest quality layer on the perceived QoE of the users in terms of quality rating. Next, we discuss how the quality switches and time on high influence the acceptance rate of the users with the provided service. At the end of the section, we summarize the findings and highlight the identified QoE influence factors.

Next, we give the relevant differences between the four user studies (C2, C3, C4 and C5) used for this evaluation. C1 was only used to verify the results from the pilot study regarding the effect of different SITI values and amplitudes and did not include a test sequence with more than one quality switch. In contrast to C2, the quality switches in campaigns C3 to C5 have high amplitudes. All campaigns, except C4, start the test sequence on the high quality level. Campaign 2 is the only campaign where no reference (i.e. zero quality switches) sequence is included for the lower quality level. For the campaigns C2 to C4, the quality switches are distributed uniform and therefore the cumulative time spent on the individual quality level differs for an even number of quality switches. For C5, all sequences have the same amount of time spent on high quality level. In the subsequent subsection, we present the results regarding the observed demographic of the test-crowd.

### 3.4.1. Demographic of the Crowd

The crowdsourcing campaigns were accompanied by a mandatory demographic questionnaire. Next, we present the demographic results from the first campaign. 150 users participated in the campaign. Out of the 150 participants, 17 gave wrong answers to the simple control questions and were excluded from the evaluation. The majority of the users (70 %) accessed the campaign's web-site from Asia, 26 % from Europe. 42 % of the participants are of age 22 - 25. The age-groups 18 - 21 and 26 - 30 were represented with 18 % each. 47 % specified Student as their occupation, followed by 32 % working in employment. 40 % of the users completed a 4-Year College, 17 % a 2-Year College and 17 % High School as their highest education. 64 % use the Internet primarily at work, whereas 36 % stated they use it primarily at home. Fixed line dominated as Internet Access technology (85 % fixed line, 15 % mobile Internet access). 97 % of the participants use the Internet on a daily basis (more than one hour per day) and 61 % visit video websites several times a day. About 31 % of the users were wearing prescription glasses. More details about the demographic of the campaign is available in the appendix (A.1).

### 3.4.2. Perception of Quality Switches

In order to judge how noticeable the quality switches are, after each presented 15 s clip, we asked the user to guess how often the video quality has changed during the playback in a range from 0 to 14 times. We implemented this question into the questionnaire of Campaign 2 and Campaign 3 to gain results for the two switching amplitudes (one quality level, two quality levels) used during the user studies.

Figure 3.6a and Figure 3.6b show the correlation between the users' guesses and

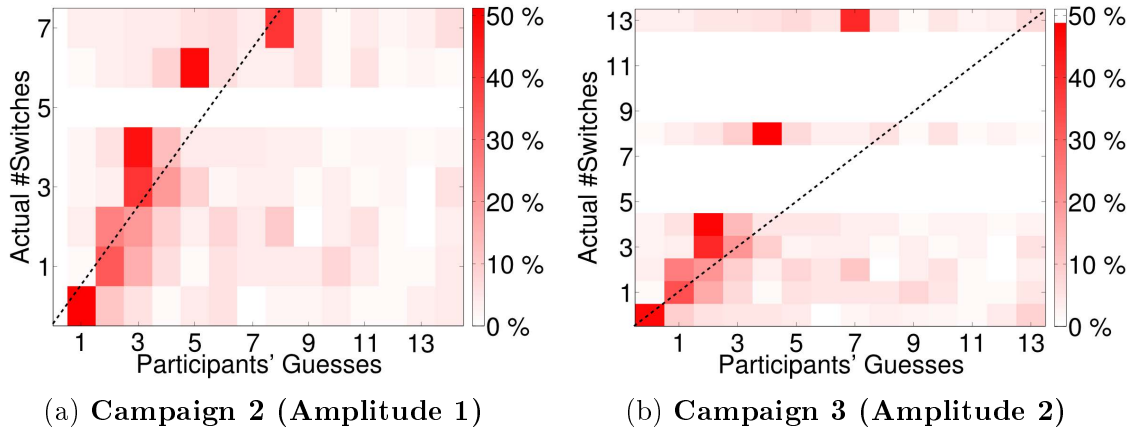


Figure 3.6.: Actual Quality Switches and User Guesses

the actual number of quality switches. The former one for the amplitude of one quality level per switch and the later one for two quality levels per switch. The dashed lines in both figures indicate where the guesses of the participants equal the actual number of quality switches for a particular test sequence. The color intensity of the square areas indicate how many users in percent guessed a certain number of switches for a particular actual number of switches.

From both figures it follows that for test sequences with no quality switches, half of the participants claimed to have noticed more than zero switches. Furthermore, 25 % (Figure 3.6a) and 29 % (Figure 3.6b) of the participants guessed that there have been greater or equal than five quality switches in the sequence. There is a general low, but significant ( $p \ll 0.05$ , i.e. a low probability to get the same result by random chance), correlation between the guesses of the participants and the actual number of switches. For the low amplitude sequences, the Pearson correlation coefficient is 0.271 and for the high amplitude sequences the coefficient is 0.247.

In summary, it can be stated that the participants were not able to accurately guess the number of quality switches. Furthermore, even in test sequences without any quality switches, the participants claimed to have noticed quality switches. However, there is a low correlation between the participant's guesses and actual the number of quality switches. This control question, asking the users to guess the number of quality switches, was removed for the campaigns C4 and C5.

### 3.4.3. Impact of Quality Switches on Perceived Video Quality

In the following, we discuss how the perceived QoE of the study participants in a test sequence is influenced by the number of quality switches and the amount of time spent on the best quality level. The results were obtained utilizing the continuous Absolute Category Rating (ACR) rating schema in a range of 0 to 1000. The subsequent figures give the results as the average quality rating. Confidence intervals are omitted for the sake of readability. On average, we observed a 95 % confidence interval length of 96 with a standard deviation of 23. The average confidence interval is indicated in the subsequent figures, labeled as *Avg. Conf. Interval.*

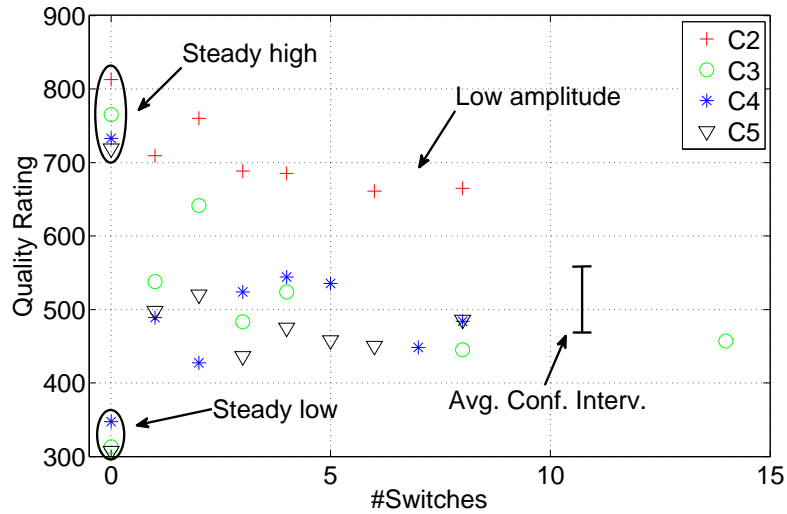


Figure 3.7.: Switches vs. Quality Rating for Campaigns 2 - 5

Figure 3.7 illustrates the influence of the number of switches on the quality rating of the participants. The two test sequences without quality switches mark the lowest (about 330 for steady low quality) and the highest observed rating (about 760 for steady high quality). For the low amplitude switches (campaign C2), there is no significant influence on the rating identifiable for a number of switches greater than two. However, the sequence with two quality switches was rated higher than the other sequences with quality switches. This can be attributed to the fact, that the two switches sequence of campaign 2 (and also of campaign 3) has a high time on high compared to the other sequences (cf. subsequent paragraph). For the high amplitude switches, a similar effect is observable. All sequences, excluding the zero and two switches sequence, do not differ in rating. However, the sequence with two switches exhibits a higher quality rating, the sequence without quality switches and steady low quality a worse rating. From the figure we conclude, that the quality rating in this study was influenced by the amplitude of the switches (a low amplitude results in better quality rating) and by the time spent on the better quality level. The results also show, that the participants preferred the sequences with quality switches instead of the sequences without quality switches but steady low quality.

Figure 3.8 highlights the influence of the time on high on the quality rating. The conclusions drawn for Figure 3.7 also apply to the figure at hand. The sequence with steady, but low quality, received the lowest average rating (about 330), the sequence with a steady high quality the highest (about 760). Furthermore, the sequences with low amplitude switches exhibit a higher quality rating than the ones with high amplitude switches. In addition to the prior conclusions, Figure 3.8 highlights the correlation between the time on high and the quality rating hinted in the previous paragraph. Based on the average quality rating, the correlation coefficient between the time on high and the quality rating is 0.82.

In the subsequent subsection, we investigate the QoE influence factors of adaptive streaming based on the acceptance rate of the different test sequences.

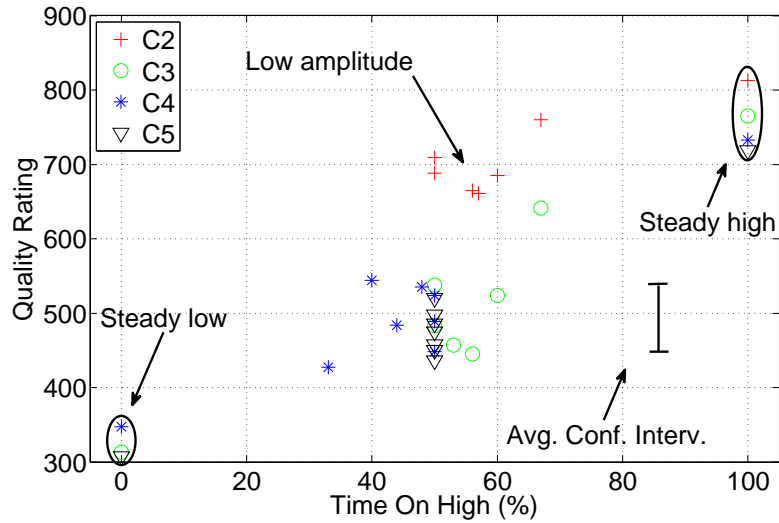


Figure 3.8.: Time On High vs. Quality Rating for Campaigns 2 - 5

### 3.4.4. Acceptance of Quality Switches

Next, we discuss how the number of quality switches in test sequence and the amount of time spent on the best quality level influences the perceived QoE of the user in terms of the acceptance rate. First, we give the acceptance rate for different number of quality switches ranging from 0 to 14. Second, we present the acceptance rate dependent on the time spent on the best quality level.

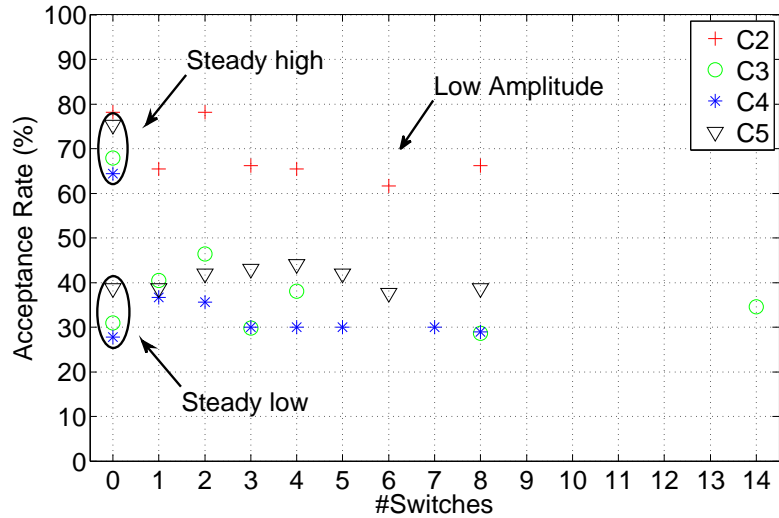


Figure 3.9.: Switches vs. Acceptance Rate for Campaigns 2 - 5

Figure 3.9 illustrates the relationship between the number of quality switches and the acceptance rate observed during the four user studies. For zero quality switches, two markers are shown for the user studies C3, C4 and C5. One for the test sequence with a steady low quality and one for the test sequence with a steady high quality. From the results presented in the figure follows, that there is no significant



correlation between the number of quality switches and the acceptance rate. For the user studies C3, C4 and C5 the acceptance rate stays low for all investigated number of quality switches and does not differ significantly from the acceptance rate of the test sequence with a steady low quality. The user study conducted with a lower amplitude (i.e. C2) also shows no influence of the number of quality switches. However, the lower amplitude results in a general higher acceptance rate equivalent to the test sequence with a steady high quality.

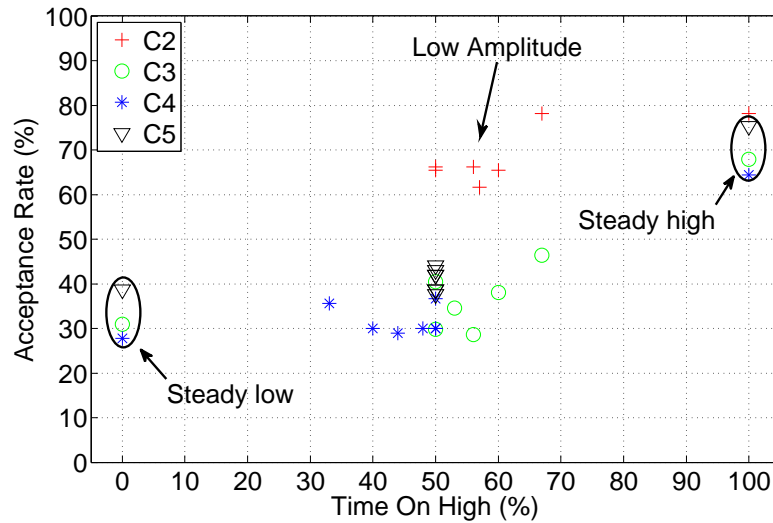


Figure 3.10.: Time On High vs. Acceptance Rate for Campaigns 2 - 5

Figure 3.10 illustrates the relationship between the time spent on the highest quality level and the acceptance rate. From the presented data follows that there is only a low correlation between the time spent on the highest quality level and the acceptance rate for the investigated range (i.e. 33% to 66% of time showing the best video quality). User study C4 does not show any significant influence regarding the time on the highest quality level. All sequences of C4 show an equally low acceptance rating. An equivalent effect is observed for C5 where the test sequences all spend half of the time on the highest quality level and the other half on the lowest. The acceptance rating is equally low for 0% and 50% time spent on the highest quality level. For a time spent on high lower than 56% of the time, C3 and C2 confirm the previous observations. However, there is an increase in the acceptance rating starting from 56% time spent on the highest quality level for C2 and C3.

In this section, we investigated how the acceptance rate is influenced by the number of quality switches and the time spent on the highest quality level. From the evaluation follows, that the number of quality switches does not influence the acceptance rate. Furthermore, we see that all sequences with more than one high amplitude switch exhibit an equally low acceptance rate equivalent to the low quality sequence without quality switches. The low amplitude switches on the other hand result in the same high acceptance rate as the test sequence with constant high quality. Regarding the percentage of time on the high quality level, all test sequences with a lower time spent on high of 56% exhibit no change in the acceptance

rate. A percentage of time spent on the higher quality equal or greater than 56% shows an increase in the acceptance rate.

### 3.4.5. Identified QoE Influence Factors

Next, we discuss and summarize the findings from the previous subsections, present the identified QoE influence factors of adaptive video streaming and the contribution to the user-centric performance evaluation of the adaptation algorithms.

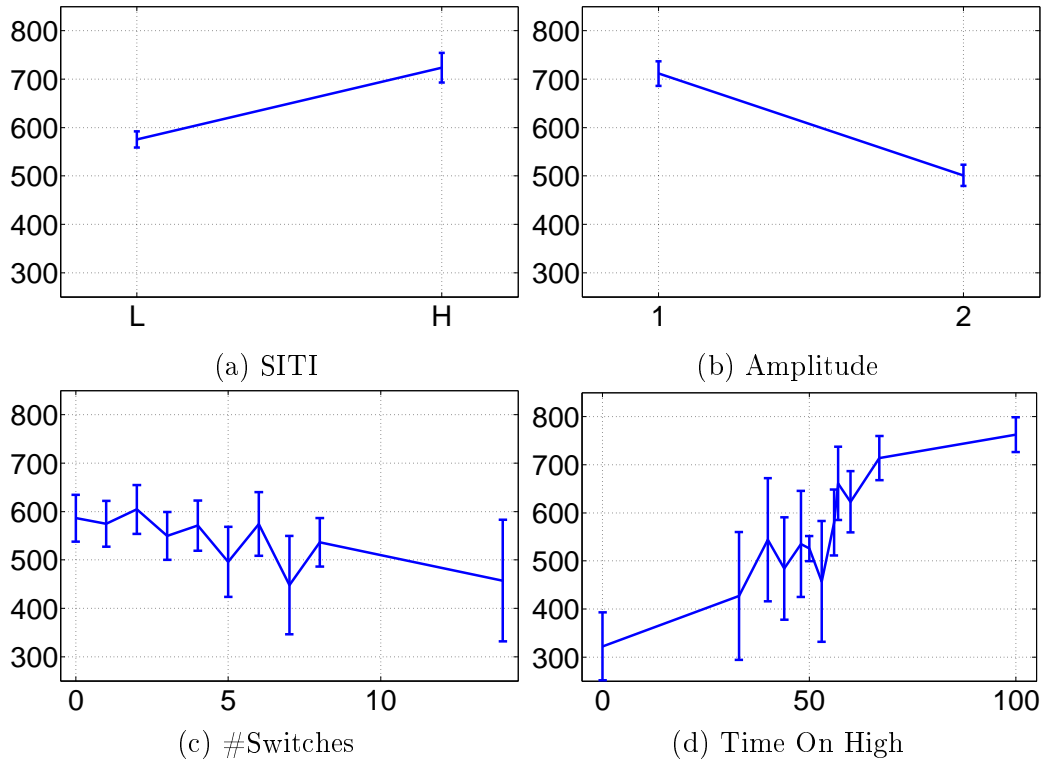


Figure 3.11.: Main effects plot for the quality rating

In the previous subsections, we investigated the influence of the number of switches, the time spent on the best quality level and the switching amplitude on the user's perceived QoE in terms of the quality rating in the range of 0 to 1000 and the acceptance rate. The results are summarized by main effect plots in displayed Figure 3.11. The findings indicate that the user's QoE is not influenced by the number of quality switches in a test sequence, for the investigated number of the quality switches of 1 to 14 per 15s sequence and considering the displayed confidence intervals, but whether there are any quality switches in the test sequence. For both metrics, the time on high and the acceptance rate, we could not observe any correlation between a number of quality switches greater than one and the ratings of the users for the observed confidence intervals. However, the sequences containing quality switches, regardless of the number of switches, were rated slightly better than the sequence with steady low quality on average and significant worse than the sequence with steady high quality. From this it follows that from the users' perspective, quality switches result in a significant lower QoE regardless of the number of switches.

The results regarding the switching amplitude confirm common assumptions. A lower switching amplitude impacts the user's QoE less than a larger amplitude. In our configuration, the lower amplitude switches showed only a slightly lower quality rating and acceptance rate than the sequence without quality switches and steady high quality. From this it follows that an adaption algorithm should keep the amplitude of the quality switches as low as possible.

The conducted studies show a correlation between the time spent on the highest quality layer and the quality rating and acceptance rate of the users. For a percentage equal or greater than 56 %, the quality rating increases with a increasing time on high percentage. For the range between 33 % and 56 % we observe a higher rating than for the 0 % time on high (steady low) sequence, but no correlation to an increasing time on high percentage. Both statements are valid for low and high amplitude switches. From the observed acceptance rates, two conclusions can be drawn. First, we observe the same increase starting from 56 % time on high percentage as for the quality rating. Second, for the user's QoE, a sequence with less than 56 % of high quality, is as acceptable as the sequence with steady low quality.

For the design of an adaptation algorithm, the following conclusions can be drawn from the crowdsourcing user studies. First, quality switches should be avoided. Second, the amplitude of the quality switches should be kept as low as possible and third, if a quality switch does not lead to a phase of higher quality longer than a previous or following low quality phase, there is no increase in the QoE of the user to expect for the investigated sequence length of 15 seconds.

## 4. An Adaptation Algorithm and Methodology for Objective Evaluation

In this chapter, we describe the proposed adaptation algorithm in detail and present the methodology used for the evaluation of the adaptation algorithms. Figure 4.1 illustrates the evaluation approach. Realistic network scenarios derived from real-world traffic traces, encoded test content and the adaptation algorithms are input parameters for an evaluation process in a test bed environment. The test-bed performs traffic shaping to simulate the network scenarios, provides the test content over HTTP and executes the DASH client and monitors the adaptation algorithms during playback. Objective metrics are deduced based on the recorded playback behavior. Afterwards, the results from the user studies are used to compare the adaptation algorithms from a user-centric point of view.

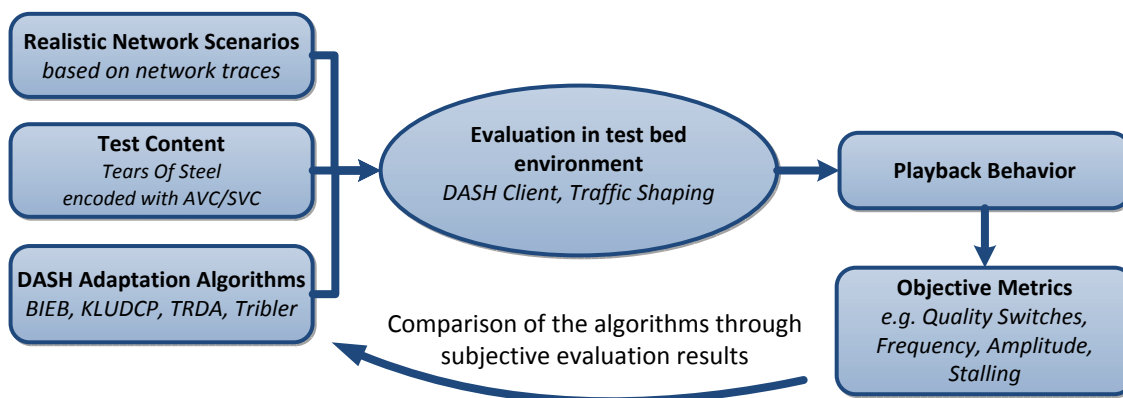


Figure 4.1.: Objective evaluation methodology overview

This chapter is structured as follows. First, we present the proposed adaptation algorithm. Second, we introduce the user- and resource-centric evaluation metrics we derive from the recorded playback behavior. Third, we highlight the characteristics of the used test content. Forth, we present the structure of the testbed used in the process of this evaluation. Afterwards, we discuss different evaluation scenarios. Evaluation scenarios allow us to answer further questions about the performance of the algorithms in real-world situations. For example, one evaluation scenario investigates the performance of the algorithms in the presence of a large file download in addition to a fluctuation network access characteristic. The section after the evaluation scenarios takes a closer look at the evaluation framework developed as

part of this thesis. We conclude this chapter by giving an overview of the evaluations performed for this thesis and their objectives. The subsequent section introduces the proposed DASH-SVC adaptation algorithm.

## 4.1. Bandwidth Independent Efficient Buffering (BIEB) Algorithm

The DASH/SVC algorithm *Bandwidth Independent Efficient Buffering* (BIEB) proposed by the author of this thesis is designed to offer a high image quality while also avoiding stalling and frequent quality switches. The algorithm does not rely on bandwidth estimations based on the current throughput and does not make assumptions about the content bitrate based on the average bitrate specified in the MPD file. It does, however, make an assumption about the relative size of segments of different representations to each other. The assumption is based on observations of encoded SVC-content and is illustrated in the following example.

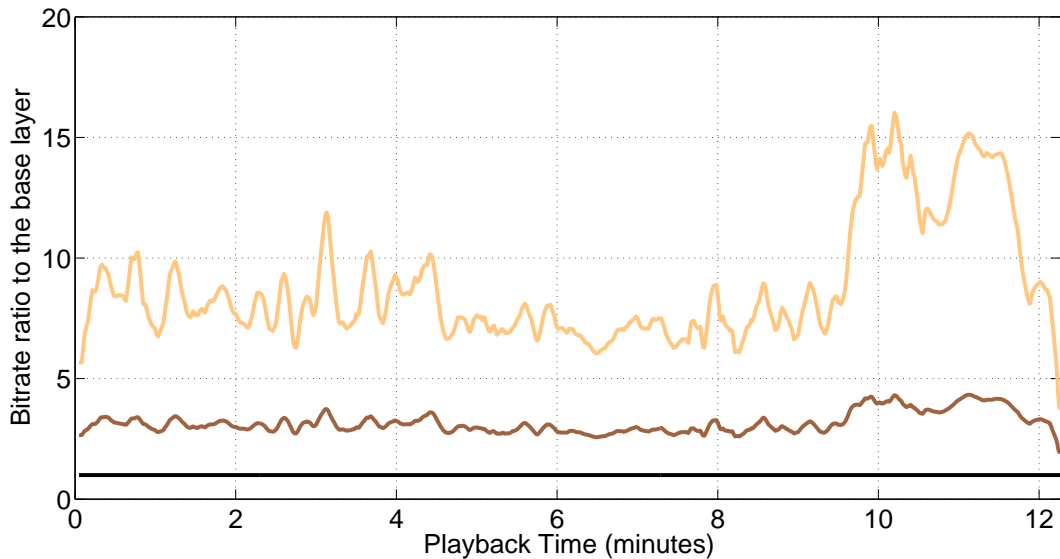


Figure 4.2.: Tears Of Steel representation bit-rates relative to the base layer

Section 4.3 introduces the open-source movie Tears Of Steel SVC-encoded with three spatial resolutions. Figure 4.5 in Section 4.3 illustrates the bit rate of the encoded bitstream separated into the three layers. Bitrate variations as shown in the example are typical for H.264-encoded content with time-varying amount of spatial and temporal information. We conclude that the average bitrate of a layer is not a reliable approximation of the bitrate for a given point in time during the playback. But, as shown in Figure 4.2, the ratio of the segment sizes of the representations relative to the base-layer stays fairly constant.

The average bitrates for each representation given by the MPD file can be used to calculate the segment size ratios used for the proposed adaptation algorithm. In detail, the following definitions are used to describe the algorithm.  $r_{avg}(i)$  is defined

as the average bitrate of representation  $i$  *without* the preceding dependent quality layers,  $br(i)$  is ratio of representation  $i$  and the base layer ( $br(i) = \frac{r_{avg}(i)}{r_{avg}(0)}$ ),  $i_{curr}$  is the highest currently selected representation (with  $i_{curr} = 0$  being the base layer, also referred to as  $i_{min}$ ),  $i_{max}$  is the highest selectable representation,  $p_{curr}$  is the segment number of the current playback position,  $\gamma$  gives the minimum buffer level in segments per each selected representation and  $\delta(i)$  returns the current buffer level in segments for representation  $i$ . A representation is called selected if at least one segment of this representation is already buffered.

The *desired buffer level*  $\beta(i)$  per each selected representation depends on the number of currently selected representations and the ratio of the segment sizes.

$$\beta(i) = \begin{cases} \gamma + br(i_{curr} - i) & \text{if } i \leq i_{max}, \\ \gamma + (i - i_{max} + 2) \cdot br(i_{max}) & \text{if } i > i_{max}. \end{cases}$$

The aim of the algorithm is to divide the currently available bandwidth evenly between the selected representations starting from the most important (i.e. the base-layer). *Evenly* is here defined by the ratio of the segment sizes as given by  $br(i)$ . The output of  $br(i)$  can also be interpreted as the *cost* of representation  $i$  where the cost of the base-layer is always 1. In the cost-model the base-layer is 'cheap' and with a growing number of selected higher, more 'expensive', representations, we also have to 'buy' more base-layer segments for the current time frame.

Figure 4.3 gives a simple example for this model. The figure shows the current buffer level in Megabytes and in number of segments per representation. In this example the segments of the representation with the resolution 1280 x 720 have each a size of six Megabytes, the segments from the next lower representation (i.e. 640 x 360) have a size of three Megabytes and the segments from the base layer have a size of one MB each. Note that all segments contain the same amount of playback time. From Figure 4.3a it follows that we have to download six segments from the base-layer, two segments from the first enhancement layer and one from the second enhancement layer to reach the desired buffer level. The resulting buffer level in number of segments is shown in Figure 4.3b. In case of one second of playback time per segment this is equivalent to the number of playback seconds buffered.

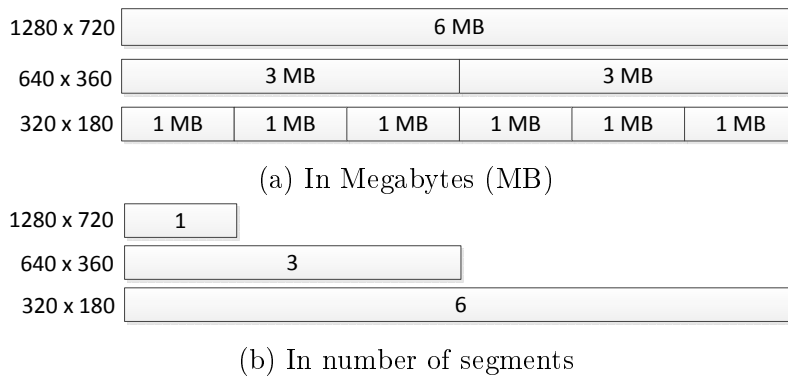


Figure 4.3.: Example buffer levels

The algorithm can be in one of three phases during the playback depending on the current buffer level. The *steady*, *growing* and *quality increase* phase. During the steady phase the algorithm tries to reach the desired buffer level for each selected representation. The growing phase begins when the algorithm has reached the desired buffer level. During the growing phase the buffer level requirements for each selected representation are increased to prepare to select the next higher representation. Note that the buffer level requirements are increased past the amount required for the next representation. This is done to inhibit any short-term throughput fluctuations from causing unwanted quality switches. The growing phase ends when the new desired buffer levels are reached. Afterwards, in the quality increase phase the algorithm selects a specific segment from the next higher representation as next segment to download and implicitly enters the steady phase again. The algorithm is called after a segment finished downloading to determine the next segment to download. Algorithm 1 defines the proposed algorithm in detail. First, the algorithm determines the number of currently selected representations by iterating through the list of representations in reverse order and stopping when the first representation with buffered segments is found. Next, the algorithm iterates through the representation starting from the base-layer and checks if all segments belonging to the steady phase are available. If one segment is missing, the algorithm stops and the segment is selected to be downloaded next. If all segments for the steady phase are available, the process is repeated with the segment requirements for the growing phase. If the segments of both phases are already available, the algorithm selects the next higher representation.

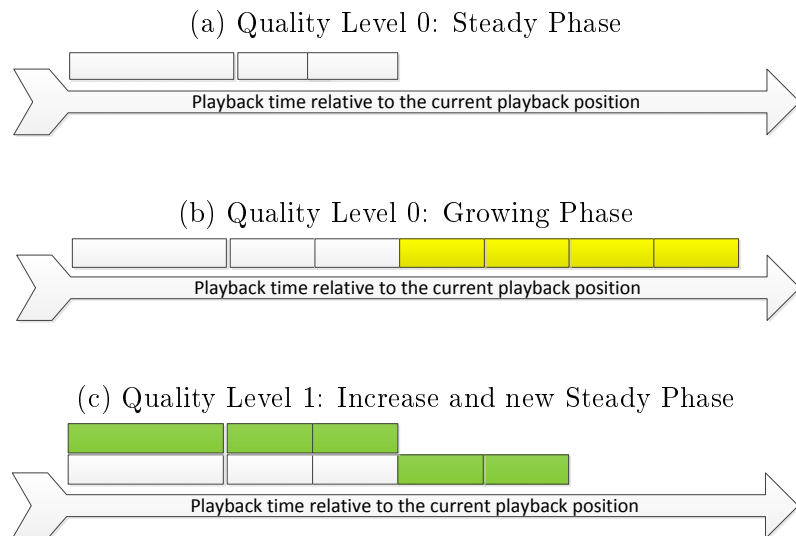


Figure 4.4.: Example Switch from Q0 to Q1

Figure 4.4 outlines the transition between two steady phases by example. At first, quality level 0 (i.e. base-layer) is selected and the minimum buffer level is available and in addition two segments. In Figure 4.4b, the algorithm enters the growing

```

i = rmin;
icurr = imax;
while  $\delta(i_{curr}) == 0$  do
  | icurr = icurr - 1;
end
i = 0; // Steady Phase
while i ≤ icurr do
  | if  $\delta(i) < \beta(i)$  then
  | | request next segment of representation i;
  | | exit;
  | end
  | i = i + 1;
end
i = 0; // Growing Phase
while i ≤ icurr do
  | if  $\delta(i) < \beta(i + 2)$  then
  | | request next segment of representation i;
  | | exit;
  | end
  | i = i + 1;
end
// Quality Increase
if i ≠ imax then
  | icurr = icurr + 1;
  | request segment pcurr +  $\gamma$  of representation i
else
  | // idle until pcurr increases
end

```

**Algorithm 1:** BIEB Adaptation Algorithm

phase, where four base-layer segments are added to the buffer. At the end of the growing phase, six base-layer segments are buffered in addition to the minimum buffer level. In Figure 4.4c, the algorithm increased the quality level by one and enters the new steady phase with two quality levels.

It has to be noted that the algorithm's desired buffer level strategy can be improved further to reach a theoretical optimum. E.g. as illustrated in Figure 4.2, the relative size ratios stay fairly constant for a long duration of the movie. However, the last two minutes exhibit a different behavior. The relative sizes of the representations to each other increase. Future work could include an estimation function of the size ratios to improve the algorithm's effectiveness in such situations. Additionally, the evaluation showed that in some situations the algorithm exhibits a *risky* segment picking behavior where the algorithm tries to download segments which are too close to the current playback position. A decrease in bandwidth (or equivalently an unexpected increase in the content bit rate) can lead to situations where the playback position already moved past the currently downloading segment



when the segment finished downloading and the segment can no longer be used and has to be discarded (referred to as *wasted bandwidth* in the evaluation). Further constraints for the segment picking can be put in place to prevent or decrease the amount of wasted bandwidth. In order to evaluate our proposed algorithm in comparison to other DASH adaptation algorithms, we also introduce a set of objective and subjective metrics in the subsequent section.

## 4.2. Utilized Evaluation Metrics

In this section, we introduce the evaluation metrics we derive from the playback behavior of the adaptation algorithms during a streaming session. We differentiate the evaluation metrics in resource-centric and user-centric metrics. Resource-centric metrics describe the behavior of an adaptation algorithm from a technical perspective (e.g. memory use, bandwidth utilization), whereas user-centric metrics characterize the algorithms by the resulting output (e.g. video quality, switching frequency).

### 4.2.1. Resource-centric Metrics

The resource-centric metrics describe how well and fair the algorithms use the available bandwidth and memory resources. Computational complexity is highly implementation-specific and is therefore not considered here.

#### Memory Use

The memory use of an adaptation algorithm is characterized by the *mean memory use* and *peak memory use* during a session. Memory use is here defined as the amount of content data buffered, not including implementation specific data structures. Peak memory is the maximum memory use of the algorithm over all conducted evaluation runs of a particular scenario.

#### Bandwidth Utilization

Bandwidth utilization is here quantified by the *ratio of the amount of downloaded data to a theoretical optimum* and the amount of data downloaded but not used (referred to as *bandwidth wasted*). The theoretical optimum is calculated from the values used by the traffic shaping process and because of its theoretical nature may not be reachable by any algorithm. Bandwidth wasted refers to situations where the adaptation algorithm downloads a segment but does not use it during the encoding process.

#### Bandwidth Fairness

*Bandwidth Fairness* is defined as the absolute and relative difference in average playback quality between two concurrent clients using the same adaptation algorithm and sharing the same Internet connection. Concurrent means that the playback session of both clients overlap during the same scenario. The start time for the second client is chosen by random between 0 and 60 seconds.

### 4.2.2. User-centric Metrics

The user-centric metrics describe the behavior of the playback during a streaming session from the perspective of an imaginary viewer.

#### Quality Switches

The quality switches are quantified by *the absolute number, the amplitude* and the *distribution of the switches* over the length of the session. The distribution of the switches is given by the inter-switching times, i.e. the length of the time intervals between two switches.

#### Playback Quality

The playback quality is characterized through the *downloaded number of segments per representation*. Derived from that, you can calculate the mean quality level, the standard deviation and objective image quality metrics like PSNR [14] and SSIM [60].

#### Stallings

Playback stalling describes the situation where the algorithm does not output any content to the video encoder. *Initial delay/stalling* describes the time interval between the download of the MPD file and the output of the first segment (i.e. the waiting time before the playback starts). Initial delay is the result of algorithms pre-buffering a specific amount of data before starting the playback. *Stalling time* is the sum of the time intervals and the number of occurrences where no content is decoded during the session, not including the initial delay. From a technical perspective, stalling during the playback is caused by buffer starvation as result of insufficient bandwidth.

## 4.3. Content Characteristics

Tears of Steel [23] is a short movie by the Blender Foundation [2] published as open-source movie with a playback length of about 12 minutes (17620 individual frames). It is freely available (Creative Commons Attribution license [16]) and features high image quality (i.e. two high resolution versions with 1080p and 720p), real actors and sophisticated visual effects in a Science-Fiction scenario.

Resolution	Average bitrate	Maximal bitrate	SSIM
320 x 180	0.294 Mbps	1.279 Mbps	0.922
640 x 360	0.949 Mbps	3.368 Mbps	0.978
1280 x 720	2.671 Mbps	10.46 Mbps	1.0

Table 4.1.: Tears Of Steel with spatial scalability

Based on the 720p version, we encoded the movie into H.264/SVC with spatial scalability using the (Joint Scalable Video Model) JSVM [19] reference software (version 9.19.15). A Group of Pictures (GOP) size of 8 frames, an IDR and Intra

period of 24 frames and QP factor of 24 was used. The encoding configuration file used for the encoding process is available in the appendix of this thesis (A.2). Table 4.1 shows the resulting average and maximal bitrates of the encoded movie with the three spatial resolutions and also the SSIM value with the highest resolution as reference quality. Doubling the resolution (i.e. 4 times the number of pixels) increases the required bitrate by approximately factor 3.

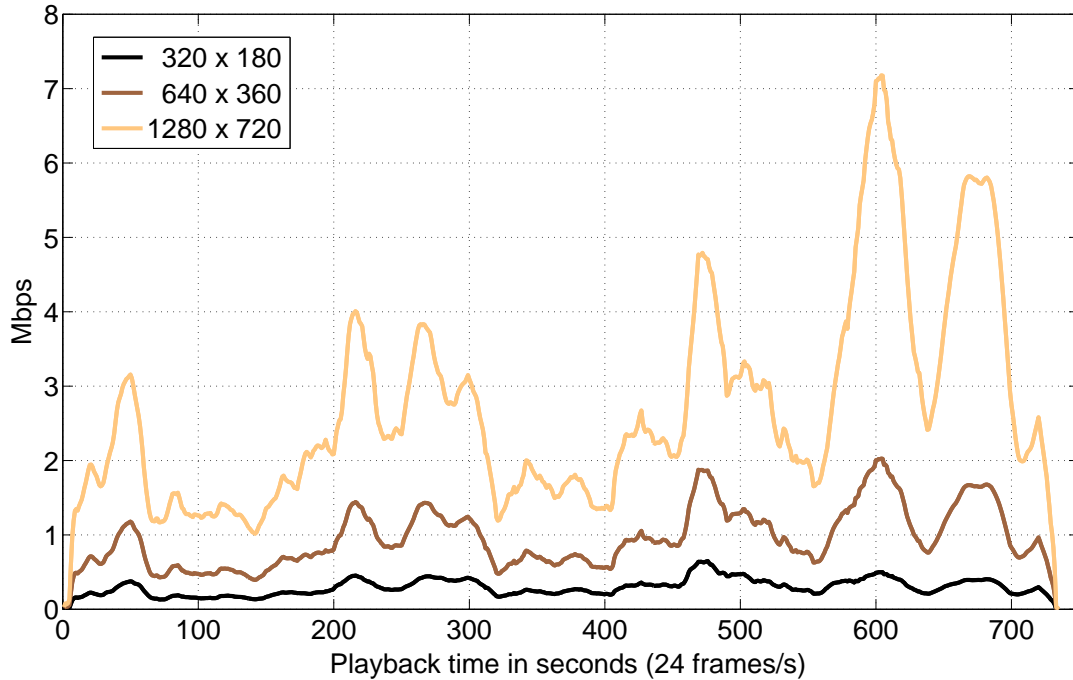


Figure 4.5.: Tears Of Steel with spatial scalability

The bitrate of the encoded movie over time is depicted by Figure 4.5. For increased readability, the bitrate is smoothed using a moving average with a window size of 10 seconds. The moving average is the reason why the maximal bitrates given in Table 4.1 differ from the bitrates illustrated by the figure. The bitrate spike at the end of the movie is due to the complex end credits. The lower spikes during the movie are caused by fast action scenes with a high amount of motion and explosions.

## 4.4. Testbed Environment

The testbed used for the experiments is illustrated in Figure 4.6. A plain HTTP server (apache2) running on a Linux host is serving the DASH content (i.e. Tears of Steel in segments, each 2s long). The traffic shaping device is running Debian Linux 6 (Squeeze) and traffic shaping is done using the Linux Advanced Routing & Traffic Control [11] framework (i.e. NetEm). It has to note, that the traffic shaping in our testbed is only applied to traffic from the HTTP server, not for data from the clients to the HTTP server. The traffic shaping process is started by uploading a script file to the shaping device which contains for each second of the evaluation run

a bandwidth and delay value. Once the evaluation process is started, the shaping device uses the supplied values to shape the traffic. After reaching the end of the script file, the shaping is started from the beginning again.

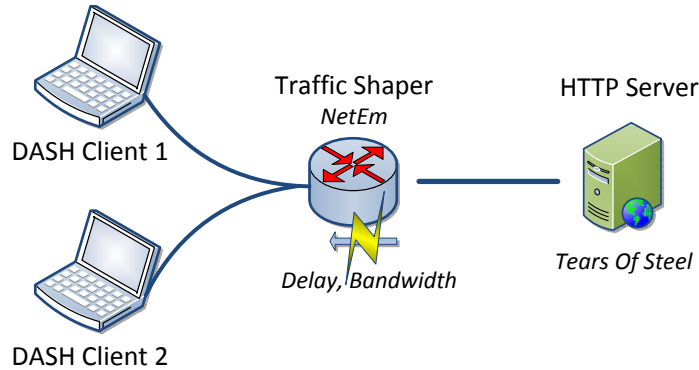


Figure 4.6.: Testbed schematic

Two DASH clients (running Kubuntu Linux 12.10) are connected to the HTTP server through the traffic shaping device by Ethernet cables and a Gbit-enabled switch. TCP Cubic [32] is used as congestion avoidance algorithm on both clients and the HTTP server. Next, we describe the different network access characteristics we utilize to shape the traffic for the evaluations.

## 4.5. Network Access Characteristics

In this thesis, network access characteristics describe the downstream transmission path between the content server and the DASH client(s) and can be either artificial (e.g. set to a constant value) or deduced from recorded network traces. Two metrics describe a network access characteristic, transmission delay and available bandwidth, both as a function of time. In the following, we first describe transmission paths with artificial constant limitations. Afterwards, we discuss the characteristics of a transmission path observed in a vehicular mobility scenario.

### 4.5.1. Constant Bandwidth Limitation

Constant bandwidth limitation is a primitive artificial network access characteristic where a specific bandwidth and delay do not change during a playback session. We define nine different constant bandwidth limitations for use in the evaluation. The limitations are derived from the test content and are multiples of the average bitrate of the base-layer ranging from two times the base-layer to ten times the base-layer. Accordingly, the different limitations are ranging from 589 Kilobits/s to 2944 Kilobits/s. The transmission delay is set to 70 ms for all limitations.

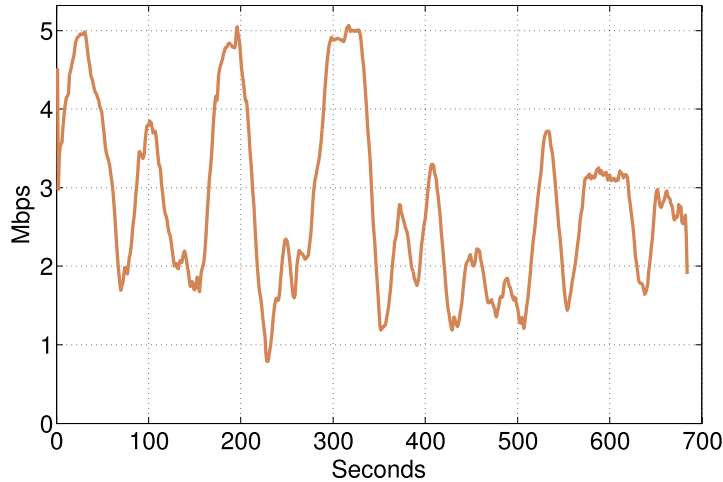


Figure 4.7.: Vehicular mobility pattern

### 4.5.2. Vehicular Mobility

The vehicular mobility network access characteristic is derived from a real-world recording. Figure 4.7 shows the measured bandwidth during a short drive on the freeway around Klagenfurt, Austria recorded by the Institute of Information Technology of the Alpen-Adria Universitaet Klagenfurt [47] using an UTMS stick. Transmission path delays were not recorded. We define a fixed transmission delay of 70 ms in the performance evaluation. The average measured throughput is 2.88 Megabits/s and the available bandwidth fluctuates rapidly in a range of about 0.8 Mbps and 5 Mbps.

## 4.6. Investigated Scenarios

The following sections describe the evaluation scenarios implemented in the objective evaluation framework. Each of the implemented scenarios is designed to mimic a specific real-world situation from an end-user's point of view. This is done by shaping the available bandwidth on the transmission path according to artificial and recorded traffic patterns and allowing event-based interferences (e.g. start of a second transmission over the same link).

### 4.6.1. Vehicular Mobility

In this scenario, we evaluate the adaptation algorithms in situation where the available bandwidth of the Internet link and the transmission delay is changing rapidly and unpredictably. The scenario is modeled with vehicular mobility in mind, where the Internet link is heavily influenced by external factors like the speed and direction of the moving vehicle or the distance to the closest base station. As network access characteristic, we utilize the recorded vehicular mobility pattern illustrated in Figure 4.7 and described in Subsection 4.5.2. The traffic pattern is played in a loop and the starting point inside the loop is randomized prior to each evaluation run. In this scenario, we only use one DASH client.

### 4.6.2. Scalability to Bandwidth

The constant bandwidth scenario is designed to evaluate how well the the algorithms' performance scale to the available bandwidth. For a constant transmission delay, the available bandwidth on the transmission path is increased by steps of 100 % of the average base-layer bitrate for each run up to the average bitrate of the highest representation (i.e. the network access characteristics described in Subsection 4.5.1). We selected the user-centric performance metrics average playback quality and switching frequency to assess the algorithms. We are interested in, whether the algorithm offer a higher playback quality for an increased bandwidth and whether the switching frequency stays on roughly the same level for different bandwidth limitations. In this scenario, we only use one DASH client.

### 4.6.3. Fairness for Two Clients

In shared living environments and households, one Internet link is generally shared among multiple client devices, where each client is constantly competing for a fair share of the bandwidth. In this scenario we evaluate the fairness between two competing DASH-clients sharing one Internet link. We assume the network to be unmanaged (i.e. best-effort delivery) and congestion avoidance is employed by the TCP implementation on the clients. Both clients are using the same adaptation algorithm and viewing the same content. For each run, the clients start the transmission time-displaced by a random amount of time between zero seconds and 60 s. As network access characteristic, we use the vehicular mobility traffic pattern.

### 4.6.4. Fairness for Cross-Traffic

In this scenario, we evaluate how the adaptation algorithms react to a competing HTTP download request on a shared Internet link. Each algorithm is evaluated twice, first with a constant available bandwidth of  $b$  and constant delay and without a competing download request and second with a available bandwidth of  $b/2$ , the same constant delay as before and a competing unlimited HTTP download request. The download request starts simultaneously with the video transmission and does not end until the video transmission has finished.

The table in Figure 4.8 summarizes the introduced evaluation scenarios by their utilized network access characteristic and the number of clients. We use the vehicular mobility traffic pattern for the fairness study and for the vehicular mobility scenario. The scenario where we evaluate how the algorithm react to different constant bandwidth limitations (i.e. Scalability to Bandwidth) and the fairness for cross-traffic scenario utilize only constant bandwidth limitations. The fairness for two-clients scenario is the only scenario where two DASH clients share the Internet connection. In the fairness for cross-traffic scenario a competing HTTP download is started concurrent to the DASH client.

Evaluation Scenario	Network Access	Number of Clients
Vehicular Mobility	Vehicular Mobility	1
Scalability to Bandwidth	Constant Bandwidth	1
Fairness for Two Clients	Vehicular Mobility	2
Fairness for Cross-Traffic	Constant Bandwidth	1 + competing HTTP download

Figure 4.8.: Evaluation Scenarios Summary

## 4.7. Evaluation Framework for HTTP DASH Measurements

In this section, we briefly discuss the DASH client implementation used in the performance evaluation of the adaptation algorithms. Since the DASH standardization was completed relatively recently, not many freely available DASH implementations exist as of today suitable for a comprehensive performance evaluation. Existing implementations were developed with different use cases in mind and therefore would have required a substantial amount of work to add additional adaptation algorithms and monitoring capabilities. As a consequence, we developed our own DASH client implementation based on freely available open-source components and included comprehensive monitoring capabilities. The following subsection gives a short introduction to the developed DASH client implementation.

### Implementation Details

The client implementation is written in C++ and utilizes the libcurl [10] library to handle HTTP requests, the pugixml [15] XML library to parse the MPD document tree and the Boost [3] libraries to increase general programming efficiency. The evaluation utilizes Matlab 2007. For a particular scenario and algorithm, three steps are performed for the evaluation. First, the framework is configured. Second, the playback is started and its behavior monitored and third, the algorithm is evaluated based on the monitored performance.

#### 1 Configuration

The DASH evaluation client is configured through command line options and JSON-based configuration files. Taken together, four input parameters are supported. *a)* The traffic pattern for the scenario, *b)* the adaptation algorithm to use, *c)* the URL of the MPD content file and if requested, *d)* any non-default parameter values for the adaptation algorithm.

#### 2 Playback & Monitoring

The behavior of the playback is exclusively dictated by the selected adaptation algorithm. The evaluation framework provides all information about the current application state (ce.g. buffer levels, current throughput) to the adaptation algorithm and the algorithm decides which segment to download next and which to send to the video decoder. This approach allows us to add new algorithms or make changes to existing without having to implement

changes to other parts of the framework. Monitoring is done by recording all information send to and send from the algorithm.

### **3 Evaluation**

The evaluation is based on the information recorded during the playback. All data objects recorded during one playback session are referred to as the data of one *run* in the terminology of the implementation. All runs with the same configuration are aggregated and referred to as a *statistics set*. In order to increase the confidence of the results, each configuration is repeated a defined number of times and therefore each statistics set contains at least two or more runs in our evaluation.



# 5. Performance Evaluation of DASH Adaptation Algorithms

In the following, we discuss the findings regarding the objective performance evaluation of the investigated adaptation algorithms. First, we present the results from the vehicular mobility scenario, a realistic scenario where the available bandwidth rapidly fluctuates during the playback. For the evaluation of this scenario we consider five metrics. This includes two QoE influence factors (switching frequency, average playback quality), as discussed in Chapter 3, as well as three resource-centric metrics, namely memory consumption, resource utilization and the amount of bandwidth wasted. Second, we take a look at how well the algorithms adapt to different bandwidth limitations in terms of the two user-centric metrics playback quality and switching frequency. To do so, we evaluate the investigated algorithms with a set of constant bandwidth limitations and compare the behavior for the different limitation settings. Afterwards, we present a fairness study where two clients using the same adaptation algorithm share a vehicular mobility Internet connection. We assess the fairness between the two clients using the metrics difference in playback quality and difference in switching frequency for both clients. Next, we discuss the behavior of the investigated algorithms for a scenario with competing cross-traffic. Specifically, we answer the question if the choice of the adaptation algorithm may influence the partition of the available bandwidth between the streaming client and a large file download. At the end of this chapter, we summarize the findings and compare the investigated adaptation algorithms.

## 5.1. Evaluation in the Vehicular Mobility Scenario

In this section, we discuss the results of the evaluation of the adaptation algorithms in the vehicular mobility scenario. We first present the findings concerning the QoE influence factors initial delay, stalling, quality switching and playback quality. Next, we show results from the resource-centric perspective, namely bandwidth utilization and memory usage. All measurements are conducted 30 times for each algorithm to get statistical significant results and in each measurement run, a random entry point was chosen for the looping traffic pattern. The results are presented with a 95% confidence interval indicated by error-bars.

### 5.1.1. Qoe Influence Factors

Playback interruptions, especially stalling during playback, but also at the beginning of a video session (i.e. initial delay) have a strong impact on the user's QoE [34]. Therefore, adaptation algorithms should be designed to minimize both, the

initial delay due to content pre-buffering, and buffer underruns due to insufficient available network bandwidth. Our findings show, that all four evaluated adaptation algorithms prevent playback stalling. Furthermore, for the initial delay, a maximum value of 2.5s was observed, which is known to have no impact on the user’s QoE [34].

In an adaptive video streaming scenario, the Quality of Experience of the streaming session as perceived by the user is influenced by two factors. The image quality and flicker effects [65]. Flicker effects are caused by the adaptation and describe multiple changes in quality over a short period of time. We specify flicker effects by their frequency in terms of switches per minute and by the length of time periods without quality switches. Image quality is here specified by time in percent of the whole playback session spend on a specific quality level. The influence of both factors on the QoE is discussed Chapter 3.

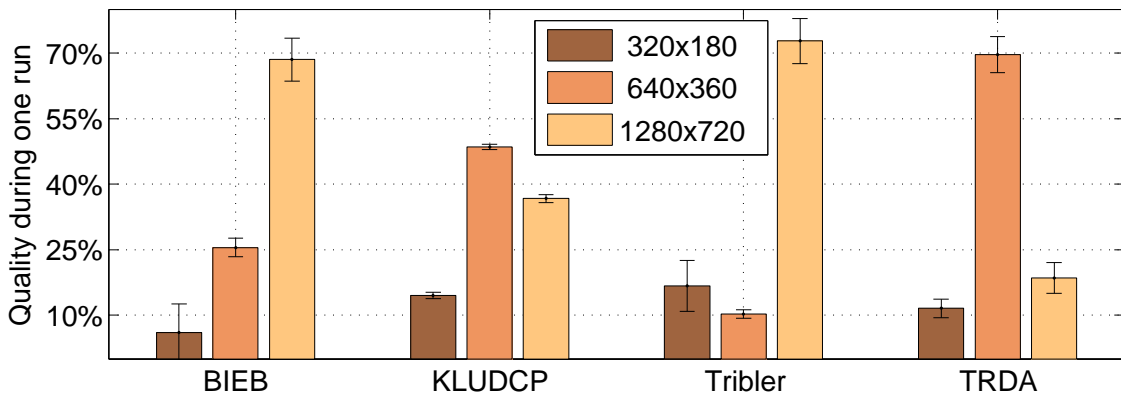


Figure 5.1.: Playback quality in time spend on the different quality levels

Figure 5.1 shows the average quality of a playback session in this scenario for the four evaluated adaptation algorithms. The figure depicts that Tribler and BIEB are able to play back more than half of the time the best quality (73%, 67%), whereas KLUDCP and TRDA show 37% and 19% of the time the best quality level, respectively. Furthermore, when comparing BIEB with Tribler, BIEB exhibits a higher percentage of the second quality level (25% to 10%). Accordingly, BIEB and Tribler both outperform KLUDCP and TRDA in terms of playback quality. When comparing BIEB with Tribler, BIEB offers an increased average quality (1.63 to 1.56, for base layer equal to 0).

Figure 5.2 and Figure 5.3 depict the quality switching behavior of the algorithms. The former one as mean and maximum quality switches per minute and the latter one as time periods of steady quality. Figure 5.2 indicates that BIEB and TRDA have a similar average quality switching frequency of 0.98 switches/min and 0.63 switches/min, respectively. The switching frequency of KLUDCP and Tribler is about 10 times higher than for BIEB and TRDA, namely 11.7 switches/min and 8.8 switches/min. Mapping this to average inter-switching times, BIEB and TRDA adapt the playback quality on average every 61.33s and 92s, respectively, whereas

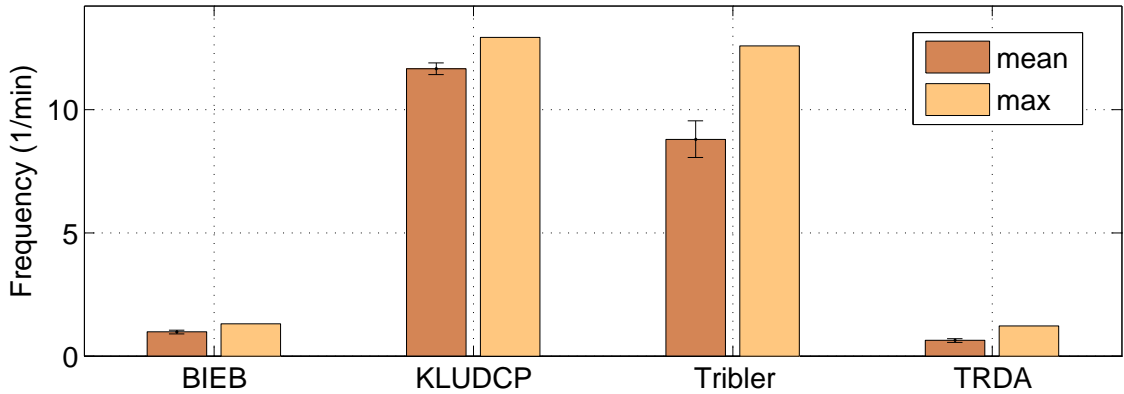


Figure 5.2.: Switching frequency

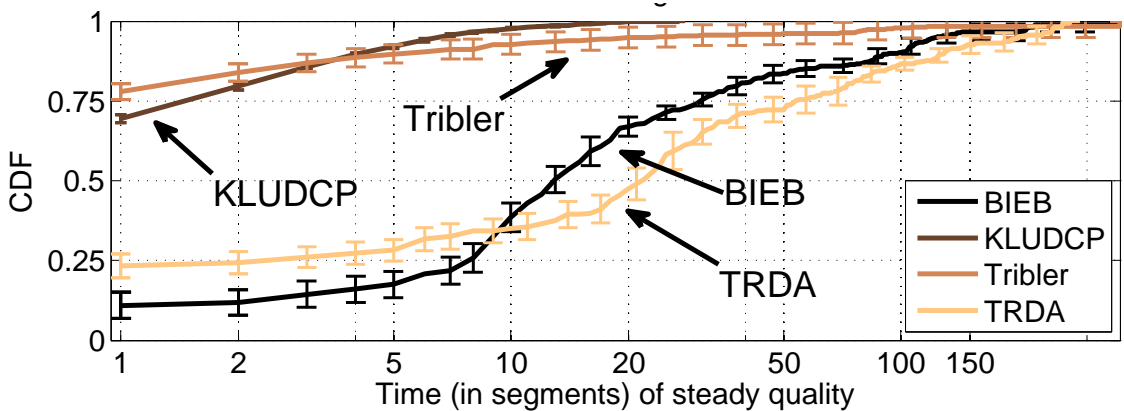


Figure 5.3.: Switching CDF

KLUDCP changes the quality every 5.15s and 6.81s, respectively. Figure 5.3 illustrates the switching behavior of the algorithms by the cumulative distribution function (CDF) of continuous segments with the same quality level. It can be seen that Tribler and KLUDCP show a similar behavior as well as BIEB and TRDA. This corresponds to the observed quality switching frequency. BIEB and TRDA can keep a quality level for a long time before they have to switch, whereas Tribler and KLUDCP have a high probability of switching the quality level after only a small number of segments. In numbers, after 10 segments (i.e. 20 seconds of playback time) of continuous playback keeping one quality level, TRDA and BIEB have a probability of about 35% for a quality switch. Whereas it is about three times more likely to switch after 10 segments of playback with KLUDCP and Tribler (95%).

In summary, it can be stated that Tribler and KLUDCP adapt the playback quality aggressively to the currently available bandwidth. This results in a high switching frequency. TRDA shows a conservative switching behavior with a very low switching frequency, but in turn can not offer a high playback quality. BIEB also shows a low switching frequency, but also offers a high playback quality. We conclude that in this scenario BIEB provides the best performance from a user-centric point of view.

### 5.1.2. Efficiency and Usage of Resources

In wireless scenarios, network and hardware resources are limited and thus an efficient usage of the available resources is important. In the following, we present the results from the resource-centric evaluation of the adaptation algorithms. Three metrics are used for the evaluation. First, the bandwidth utilization. Second, the amount of wasted bandwidth and third, the memory consumption during the playback in terms of buffered segment data.

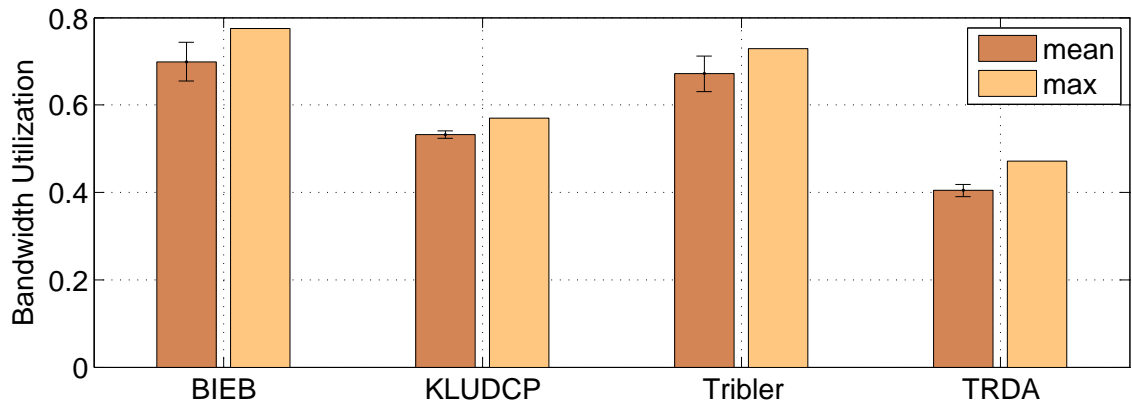


Figure 5.4.: Bandwidth utilization relative to a theoretical maximum

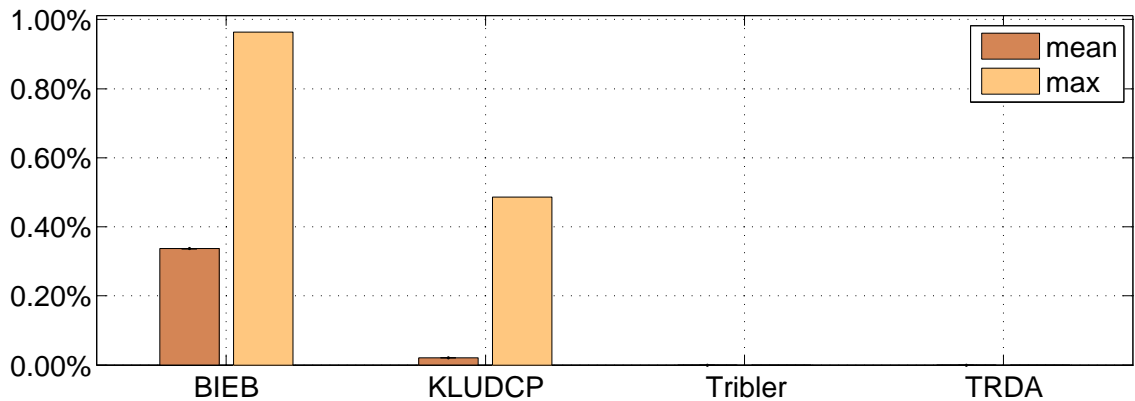


Figure 5.5.: Bandwidth wasted in percentage of movie file size on highest quality

Figure 5.4 illustrates the bandwidth utilization of the four adaptation algorithms on a scale of 0 to 1 (i.e. 0 % to 100 % of the theoretical maximum). On average, BIEB provides the highest utilization with 70 %, closely followed by Tribler with 67 %. KLUDCP and TRDA offer a lower network resource utilization, with KLUDCP using 53 % and TRDA 40 % of the available resources.

Adaptation algorithms may choose to download segments which are not used during the encoding process. Therefore, bandwidth utilization alone does not reflect the overall network efficiency. The average amount of data wasted by the algorithms is presented in Figure 5.5, given as the percentage of the file size of the content

including all higher layer segments. In 30 runs, the adaptation strategy of Tribler and TRDA did not result in any discarded segments during the encoding process. The amount of data discarded by KLUDCP is considered insignificant. BIEB discards segments in 66 % of the runs, with an average wasted data of 0.8 Mbyte per run.

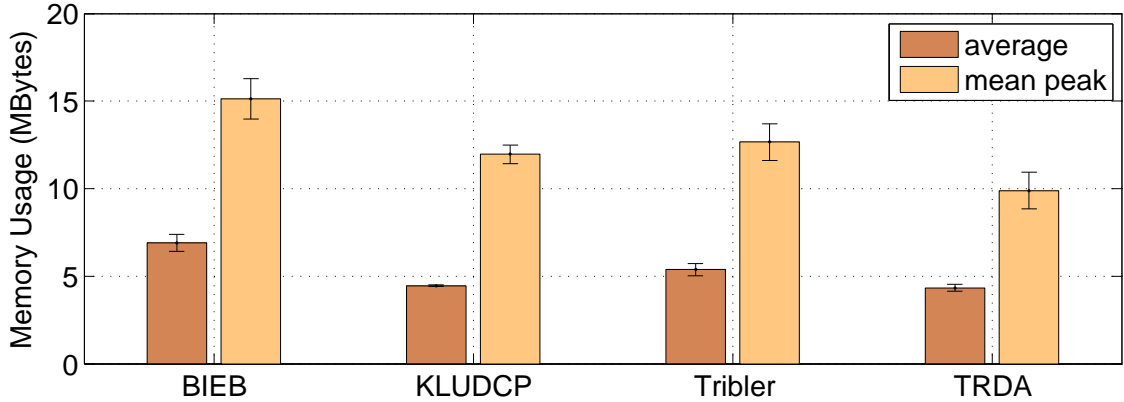


Figure 5.6.: Memory usage

Figure 5.6 illustrates the average and mean peak memory use of the algorithms. The peak memory consumption is the maximal amount of segment data the algorithm buffers at a point in time during one run. It can be interpreted as the minimum memory requirement for a mobile viewing device to support the playback of the test content in this scenario. The figure also indicates that none of the four adaptation algorithms uses an extensive amount of memory. BIEB has the highest peak memory consumption of 15.1 MBytes (i.e. 6.6 % of the whole test content), which is an insignificant amount of data compared to today’s mobile device’s memory resources. BIEB is followed by Tribler with a memory consumption of 12.7 MBytes and KLUDCP with 12 MBytes. TRDA has the lowest peak memory consumption (9.9 MBytes) of the four investigated adaptation algorithms.

In summary, it can be said that BIEB and Tribler manage to utilize a higher percentage of the available network resources in contrast to TRDA and KLUDCP. BIEB has to discard some of the downloaded data, but additional segment picking constrains may decrease the amount of wasted data. None of the algorithms exhibits an extensive use of memory for the investigated scenario and test content.

## 5.2. Playback Quality and Switching Scalability

In this chapter, we evaluate how well the adaptation algorithms scale to the available bandwidth in terms of playback quality and if the switching frequency behavior is the same for different limitations. To do so, we choose a number of bandwidth limitations evenly distributed between two times and ten times the average bit-rate of the base layer and perform ten evaluation runs for each algorithm and bandwidth limitation. Confidence intervals were observed to be small for constant bandwidth limitations and therefore omitted here for the sake of readability. The following figures show the results for two selected performance metrics for each algorithm and

limitation as the mean of the results gathered from the ten runs. First, we present the average playback quality for the different limitations. Afterwards, we display the results regarding the average switching frequency.

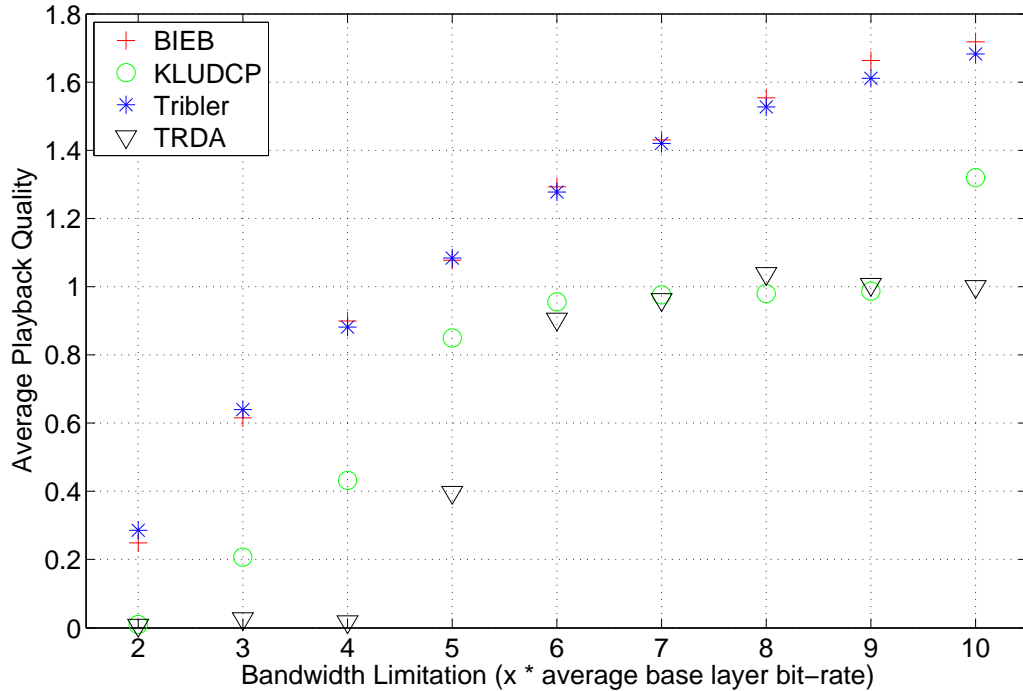


Figure 5.7.: Average playback quality

Figure 5.7 indicates the average playback quality for different bandwidth limitations. The playback quality is here defined in a range of zero to two, with zero being the quality level 0 (i.e. the base layer) and 2 being the second enhancement layer of the test content. The figure highlights that for BIEB and Tribler, an increase in bandwidth always leads to an increase in playback quality and both can offer approximately the same average playback quality. KLUDCP and TRDA however show a different behavior. TRDA only shows the base layer up to a bandwidth limitation of five times the bitrate of the base layer. For five and six times the base bitrate, the average playback quality increases and remains roughly constant for all higher evaluated limitations. KLUDCP increases the average playback quality for three, four, five and ten times the base bitrate. For six to nine, the playback quality remains roughly constant. We can conclude, that BIEB and Tribler always increase the average playback quality for the the evaluated limitations. Whereas TRDA and KLUDCP exhibit limitations where the average playback quality does not improve with increasing available bandwidth.

Figure 5.8 shows the average switching frequency for the different evaluated bandwidth limitations. The figures shows that BIEB and TRDA keep the switching frequency low for all evaluated limitations, whereas KLUDCP and Tribler exhibit an unstable behavior. Tribler only provides a low switching frequency for six times the base bitrate and KLUDCP for two, seven, eight and nine times. To put it in a

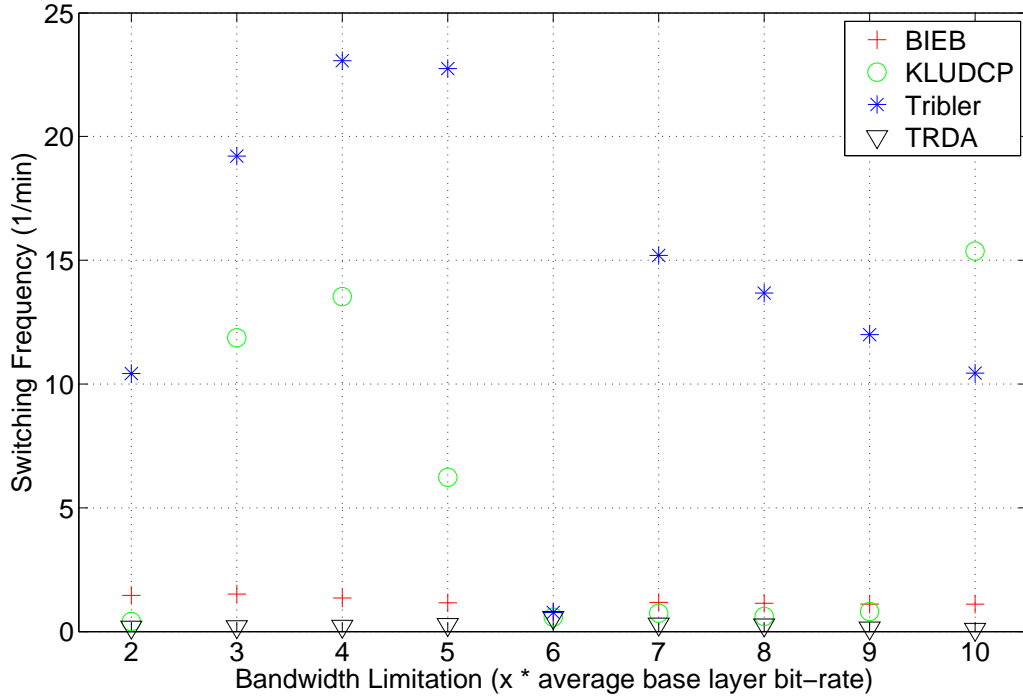


Figure 5.8.: Average switching frequency

nutshell, BIEB and TRDA outperform KLUDCP and Tribler in terms of stability of the switching frequency for the evaluated limitations. We come to the conclusion that the oscillation of the KLUDCP algorithm is due to the utilized bandwidth estimator. Hence, it may cause an increase in the switching frequency if the available bandwidth is close to the average bitrate of a quality level.

The evaluation indicates, that the proposed algorithm BIEB outperforms the other three adaptation algorithms in terms of stability of playback quality and switching frequency for the evaluated bandwidth limitations. For BIEB and Tribler, the average playback quality scales well to the available bandwidth. However, Tribler exhibits an unstable behavior for the switching frequency. TRDA offers a stable switching frequency, but provides a low average playback quality which is also does not scale well to the available bandwidth.

### 5.3. Playback Quality and Switching Fairness

In the following, we discuss the results of the concurrent clients experiment where we evaluate the fairness of the adaptation algorithms in a competitive setting for the two user-centric metrics playback quality and switching frequency. In this scenario two clients share one Internet connection and concurrently watch the same content utilizing the same adaptation algorithm. As network characteristic for the Internet connection we use the vehicular mobility traffic pattern (average bandwidth 2.88 Megabits/s). During each run, the two clients start the video playback time-displaced by a random number of seconds between 0 s and 60 s. As evaluation metrics in each run, we use the difference in number of quality switches and difference of the

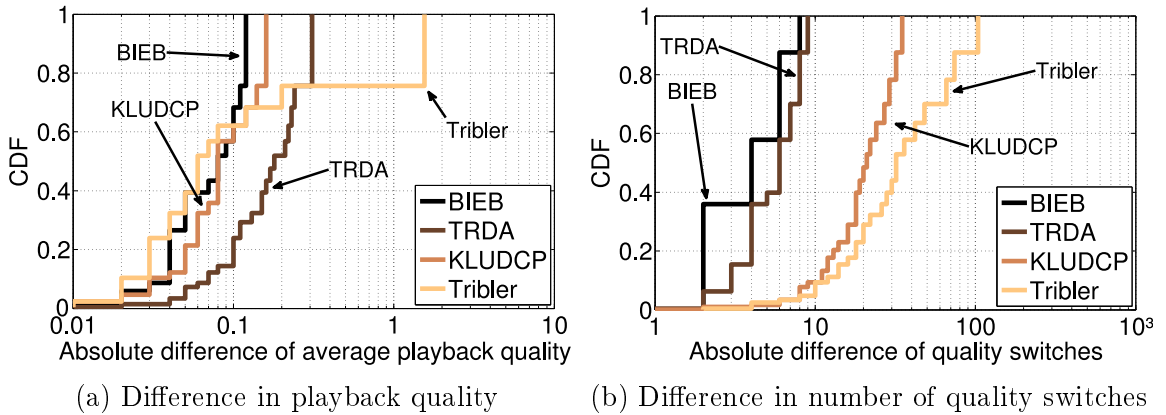


Figure 5.9.: Difference between two concurrent clients

average playback quality of the two clients.

Figure 5.9a shows the cumulative distribution function (CDF) of the absolute differences of the average playback qualities between the two clients for all runs. The difference in average playback quality shown on the bottom axis can range between 0 (i.e. no difference in playback quality) and 2 (i.e. one client only shows the base layer, the other client only the second enhancement layer). The figure depicts that BIEB and KLUDCP exhibit a similar fairness between the two clients. The difference in average playback quality is 0.12 or less (6 % of the maximal possible difference) and 0.16 or less (8 %) when using BIEB and KLUDCP, respectively. Tribler exhibits a similar fairness as BIEB and KLUDCP in 70 % of the evaluation runs, but the maximal difference can be as high as 1.55 (78 %). For TRDA the maximal observed difference is 0.31 (15.5 %). In summary, it can be stated that BIEB and KLUDCP exhibit a high fairness compared to TRDA and Tribler in this scenario in terms of average playback quality. On average, Tribler shows a similar fairness but we observe runs with highly unfair behavior. TRDA exhibits a more unfair behavior than the other three investigated adaptation algorithms.

Figure 5.9b shows the CDF of the absolute differences of the number of quality switches per run. In this scenario, for BIEB and TRDA, the number of quality switches between the concurrent clients does not differ by more than eight and nine switches (i.e. a difference of 0.66 and 0.74 switches per minute), respectively. For KLUDCP and Tribler, the probability to observe a difference of eight or less is 7.7 % and 4.7 % and the difference is always 35 and 104 or less (i.e. a difference of 2.87 and 8.52 per minute), respectively. From this it follows, that the adaptation algorithms BIEB and TRDA show a high fairness between two concurrent clients in terms of number of quality switches. Whereas, when using the adaptation algorithms KLUDCP and Tribler, the probability of one client exhibiting a significant larger number of quality switches than the other, is high.

We can conclude that BIEB is the only evaluated adaptation algorithm offering a high fairness in terms of both metrics, playback quality and number of quality switches. KLUDCP shows a high fairness regarding the average playback quality, but performs poorly in terms of difference in number of quality switches. TRDA offers a high fairness regarding the quality switches, but exhibits a low fairness for



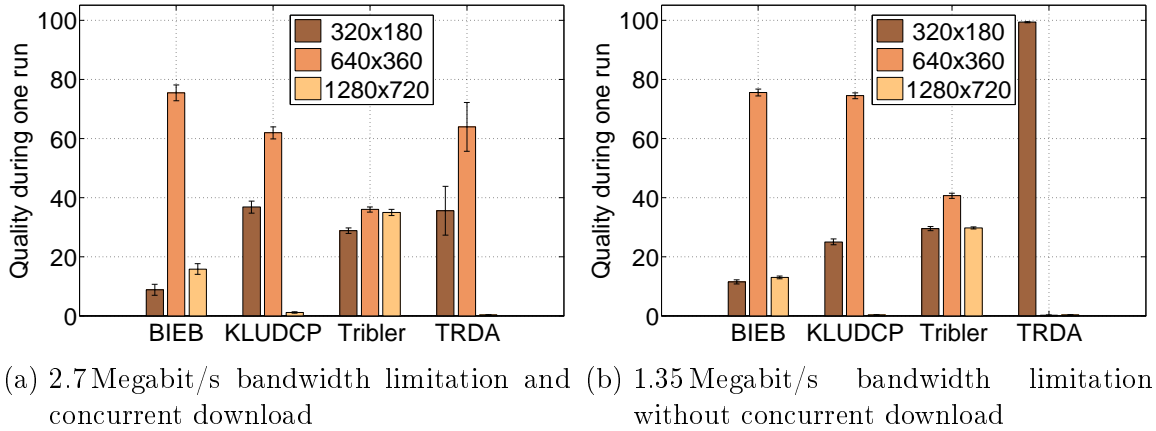


Figure 5.10.: Difference in playback quality for competing download traffic

the average playback quality. Tribler depicts highly unfair behavior in terms of number of quality switches. However, the fairness regarding the average playback quality is high in about 80 % of the observed runs.

## 5.4. Bandwidth Fairness for Cross-Traffic

In this scenario, we answer the question if the choice of the algorithm effects the distribution of the available bandwidth between a concurrent file download and a video playback session. In order to do so, we deploy each algorithm twice per experiment. First, we set the available bandwidth to a fixed amount of 2.7 Megabit/s and start a concurrent large file download simultaneously to the playback session. For the second run, we cut the available bandwidth in half and start the playback session without a concurrent HTTP file download. We repeat the experiment 30 times for each algorithm to gain statistically significant results. We use the metrics *playback quality per run*, the *number of quality switches* and the *distribution of the bandwidth* to assess the effect of the concurrent download on the playback session.

Figure 5.10 shows the observed playback quality for (a) the 2.7 Megabit/s bandwidth limitation with the concurrent download and (b) the 1.35 Megabit/s bandwidth limitation without the concurrent download. The figures show that in both cases BIEB and Tribler offer a similar playback quality, whereas KLUDCP and TRDA exhibit a higher playback quality for (a). For TRDA the difference in playback quality between (a) and (b) is the largest of all four algorithms. In (a), TRDA is able to download more segments from the first enhancement layer than from the base layer, whereas in (b), TRDA only downloads the base layer for the whole playback session. From this it follows, that in terms of playback quality, BIEB and Tribler are not effected by a simultaneous large file download, whereas KLUDCP and TRDA exhibit a significantly different playback behavior.

The evaluation regarding the change in switching frequency shows that the switching behavior of the algorithms is not considerably effected by the file download in this scenario. Tribler exhibits the most significant effect of the four algorithms

with an absolute change of the switching frequency from 23.2 switches per minute for the 1.35 Megabit/s bandwidth limitation to 21.4 switches per minute for the 2.7 Megabit/s bandwidth limitation with the concurrent download. BIEB displays the least significant effect with a change from 1.3 to 1.1 switches per minute. In terms of bandwidth distribution as the ratio between download traffic and video playback traffic, the algorithms can be divided up in two groups. BIEB and Tribler provide a fair bandwidth distribution of 1.04 and 0.99, respectively, whereas KLUDCP and TRDA get suppressed by the file download (1.77 and 1.92, respectively).

In summary, it can be stated that BIEB is the least effected adaptation algorithms of the four investigated algorithms in this scenario. For all three evaluated metrics, change in playback quality, change in switching frequency and bandwidth distribution, the algorithm does not show a significant difference between the two deployed bandwidth limitations. Tribler also displays no significant change in playback quality and bandwidth distribution, but the algorithm exhibits the largest difference for the switching frequency. However, compared to the absolute number of quality switches, that change can also be considered as negligible. KLUDCP and TRDA show a significant change for the observed playback quality. Regarding the bandwidth distribution, these algorithms are not able to share the available bandwidth evenly with the cross-traffic. They only utilize less than half of the available network resources. Future work on this topic should include additional bandwidth limitation patterns to allow for a more general evaluation of the behavior of the algorithms in the concurrent download traffic scenario.

## 5.5. Comparison of the Investigated Algorithms

Next, we summarize the findings of the objective performance evaluation and compare the four investigated adaptation algorithms. The findings are qualitatively summarized in Table 5.1. The grading scale high, medium and low refers to the measured values for each evaluated metric.

In the vehicular mobility scenario, we evaluated the four adaptation algorithms in a challenging scenario where the available bandwidth heavily fluctuates. The results show, that the SVC-based adaptation algorithms BIEB and Tribler can offer a high playback quality and are able to utilize most of the available bandwidth in this difficult scenario. However, the use of Tribler results in a high number quality switches during the playback. In contrast to Tribler, the conservative approach of TRDA can keep the number of quality switches very low, but to do so, it does not utilizes the available bandwidth well and hardly switches to the next higher playback quality. This results in a low bandwidth utilization and a lower playback quality. KLUDCP shows a similar low playback quality and bandwidth utilization, but in contrast to TRDA, the switching frequency is high. In terms of memory requirements, all four evaluated algorithms show a low consumption. It has to be noted that BIEB discards some segments during the playback (referred to as bandwidth wasted), but compared to the size of the content, the amount of discarded data is neglectable. From this follows, that BIEB outperforms the other three algorithm in this scenario.

In addition to the vehicular mobility scenario, we run the algorithms in three

	BIEB	KLUDCP	Tribler	TRDA
Using SVC or AVC	SVC	AVC	SVC	AVC
Vehicular Mobility Scenario				
<b>Playback quality</b> (avg. quality, base=0)	high 1.63	med 1.22	high 1.56	med 1.07
<b>Quality switching frequency</b> (avg. switches per minute)	low 0.98	high 11.7	high 8.79	low 0.63
<b>Bandwidth utilization</b> (avg. bandwidth utilization)	high 70%	med. 53%	high 67%	low 40%
<b>Wasted data</b> (avg. wasted data)	med 0.33%	low 0.02%	low 0%	low 0%
<b>Memory consumption</b> (avg. memory consumption, MBytes)	low 6.9	low 4.46	low 5.38	low 4.33
Artificial Scenarios				
<b>Switching Consistency</b>	high	med.	low	high
<b>Quality Scalability</b>	high	med.	high	low
<b>Fairness Playback quality</b>	high	high	high/med.	med.
<b>Fairness Nr. of quality switches</b>	high	low	low	high
<b>Bandwidth Fairness</b> ( <i>ratio Download/DASH</i> )	high <i>1.04</i>	low <i>1.77</i>	high <i>0.99</i>	low <i>1.93</i>

Table 5.1.: Comparison of the investigated algorithms

additional testbed configurations to assess the scalability and fairness of the algorithms. We first evaluated how consistent the behavior of the algorithms is for different constant bandwidth limitations in terms of switching frequency. We also assessed, how the resulting playback quality correlates to the deployed bandwidth limitations. Next, we evaluated the fairness of the algorithms in experiments where two concurrent clients were competing for the available bandwidth. As metrics we used the difference in playback quality and number of quality switches between the two clients. Following, we assessed how the algorithms react to a competing large file download request. Again, using the metrics playback quality and quality switches. The results show, that BIEB and TRDA manage to keep the switching frequency low for all tested constant bandwidth limitations, whereas the switching behavior of KLUDCP and Tribler is dependent on the available bandwidth. From this it follows, that the two algorithms BIEB and TRDA show a highly consistent behavior across the series of tested bandwidth limitations in contrast to KLUDCP and Tribler. For the playback quality scalability, we observe that for BIEB and Tribler an increase of available bandwidth is equivalent to an increase in average playback quality for the evaluated limitations. Whereas for KLUDCP and TRDA, the playback quality remains roughly constant across most limitations. In terms of fairness between two clients sharing the same Internet connection and deploying the same adaptation algorithm, BIEB is the only algorithm which can offer a high fairness for the quality of the playback and number of quality switches. KLUDCP and Tribler also offer a high fairness for the playback quality, but both exhibit an unfairness

regarding the number of quality switches. TRDA, as does BIEB, also exhibits a fair behavior for the number of quality switches, but only shows a medium fairness for the playback quality. In the fourth scenario, we evaluated if the algorithms show a different behavior when confronted with a simultaneous large file download request and how well the available bandwidth is distributed between the video playback and the file download. The results show, that the two adaptation algorithms which base their adaptation decisions on estimations of the available bandwidth, KLUDCP and TRDA, exhibit a different behavior for the experiments with the concurrent file download enabled compared to experiments without the file download. BIEB and Tribler do not show any influence of the file download on the used metrics in the experiments. In terms of bandwidth distribution, BIEB and Tribler are not suppressed by the large file download. The video playback and the download request share the available bandwidth evenly. The use of KLUDCP and TRDA results in a highly unfair distribution of the bandwidth where the file download uses a larger percentage of the available bandwidth than the video playback.

In summary, it can be stated that the proposed BIEB algorithm outperforms the other adaptation algorithms in the vehicular mobility scenario and in terms of scalability and fairness. TRDA shows a very conservative behavior which results in a low switching frequency, but also low playback quality. Tribler is very aggressive and can offer a high playback quality, but at the cost of a high switching frequency. KLUDCP can not show a satisfying behavior for the evaluated metrics and configurations.

## 6. Conclusion and Outlook

In 2012, 60 % of the global Internet IP traffic was generated by video streaming and predictions show an increase of its traffic share to 73 % by 2017 [4]. Additionally, more and more traffic originates from mobile devices, with studies predicting mobile traffic to overtake wired traffic by 2017. Of the mobile data traffic in 2012, already roughly half of the traffic was generated by video streaming. This is due to the world-wide adaption of mobile devices like smartphones and tablets which create a demand for content to be available on all of the user's Internet video-enabled devices.

But video streaming is costly for content providers in terms of bandwidth, storage and traffic. In recent years, the HTTP protocol on top of the TCP transport protocol gained popularity among video content providers as an efficient way to deliver none real-time content (i.e. pre-encoded Video-on-demand content) to their customers. With the increasing use of wireless and mobile devices for video streaming, content providers have to offer robust solutions which can adapt the video playback to the viewing environment. Dynamic Adaptive Streaming over HTTP (MPEG-DASH) was standardized 2012 on top of HTTP/TCP to allow for client-side adaptation of the video playback. However, the adaptation process is out of scope of the MPEG-DASH standard and the design or choice of an adaptation strategy is left to the client implementation. But there is little knowledge about the relationship between adaptation strategy and resulting playback behavior. Additionally, it is uncertain how the perceived Quality of Experience (QoE) of the user is influenced by the adaptation process. Furthermore, this thesis shows that none of the investigated MPEG-DASH adaptation algorithms taken from the literature can offer satisfying results for all of the evaluated metrics.

The first contribution of this thesis is the identification of influence factors of adaptive video streaming on the user's QoE. We conducted a laboratory study and multiple crowdsourcing campaigns to gain a better understanding of the influence of the adaptation process on the user's perceived quality. The second contribution of this thesis is the design of a user-centric adaptation algorithm (BIEB). The design of the algorithm aims for a high average playback quality while also avoiding the identified negative influence factors of the user's QoE. The third contribution is the user- and resource-centric comparison of the proposed algorithm to three existing algorithms (TRDA, KLUDCP, Tribler) based on the identified influence factors. In order to do this, we designed and implemented a test bed environment where the algorithms were evaluated utilizing realistic network scenarios. Additionally, we evaluated the fairness of the algorithms regarding two scenarios where a competing DASH client and HTTP cross-traffic share the Internet connection with the video streaming. We also assessed in another scenario how the performance of the algorithms changes for different constant bandwidth limitations. The results of these additional scenarios comprise the fourth contribution of this thesis.

Our findings regarding the influence factors of adaptive video streaming indicate, that the occurrence of quality switches, the amplitude of the switches and the time spend on the different quality levels have a significant influence on the perceived QoE and the acceptance rate of the service. For the design of an adaptation algorithm, the following conclusions can be drawn from the findings. First, quality switches should be avoided. Second, the amplitude of the quality switches should be kept as low as possible. Third, for the investigated sequence length of 15 seconds, switching the quality has only a positive influence on the QoE if there is a subsequent phase of higher quality longer than half of the sequence length.

The performance evaluation of the four investigated algorithms shows that the playback behavior of the streaming session is highly dependent on the utilized adaptation algorithm. In the vehicular mobility scenario with rapidly varying bandwidth, only BIEB and TRDA can offer a low switching frequency. However, TRDA's conservative switching behavior leads to a medium average playback quality, whereas BIEB presents a high average quality. Tribler and KLUDCP in turn exhibit a high switching frequency, but of those two, only Tribler can offer a high average quality. In terms of bandwidth utilization, BIEB and Tribler can utilize a high percentage (70 %) of the available bandwidth, whereas KLUDCP can only use roughly half (53 %) and TRDA 40 % of the available resources. BIEB inhibits a risky segment picking behavior and on average, discards 0.33 % of the test content in full quality. Constrains for the segment picking may reduce or avoid this behavior. Regarding the performance of the algorithms for different constant bandwidth limitations, we found that only BIEB and TRDA offer a low switching frequency for all investigated limitations, whereas KLUDCP and Tribler show a inconsistent behavior. The same applies for the average playback quality, where BIEB and TRDA always exhibit an increase of average quality for an increase in available bandwidth. For the fairness study, we observe that only BIEB can offer a high fairness for the two metrics playback quality and number of quality switches between two concurrent clients. From the objective performance evaluations follows, that BIEB outperforms the other three investigated algorithms for the tested configurations.

This thesis represents a first step towards Quality of Experience (QoE) management of adaptive video streaming in wired and mobile scenarios. The knowledge about the influence of the adaptation process on the user's perceived Quality of Experience and the comparison of the algorithms can help video content provider to assess and improve their service performance. Future work on this topic should investigate additional test content, scenarios with more than two clients sharing one Internet connection and the fairness between different adaptation algorithms.

# A. Appendix

The appendix of this thesis contains the relevant information about the user studies. Additionally, the section A.2 gives a list of the files and image sequences stored on the data medium of this thesis. Next, we present the demographics of the first crowdsourcing user study campaign.

## A.1. Crowd-sourcing Demographic Campaign C1

In the following, we present the demographic results of the first crowdsourcing campaign. The questionnaire utilized to gain the results is available in A.5. Table A.1 and Table A.2 summarize the results of the questionnaire.

Crowdsourcing Demographic (Part I)		
Sex	Male	86 %
	Female	14 %
Continent	Africa	3 %
	Asia	69.2 %
	Australia	0.75 %
	Europe	25.6 %
	North America	1.5 %
	South America	0 %
Average Internet Usage per day	Less than 1 hour/day	3 %
	1 - 5 hours/day	33.8 %
	5 - 7 hours/day	25.6 %
	7 - 10 hours/day	20.3 %
	more than 10 hours/day	17.3 %
Occupation	Working	32.3 %
	Unemployed	13.5 %
	Student	46.6 %
	Apprenticeship	3.8 %
	Pensioner	0.75 %
	Home-keeper	3 %
Age	18 - 21	18 %
	22 - 25	42.1 %
	26 - 30	18 %
	31 - 40	16.5 %
	41 - 50	4.5 %
	51 - 60	0.75 %
	61+	0 %

Table A.1.: Crowdsourcing Campaign C1 demographics (Part 1)

Crowdsourcing Demographic (Part 2)		
Education	< High School	3.0 %
	High School/GED	17.3 %
	Some College	8.3 %
	2-Year College	17.3 %
	4-Year College	39.8 %
	Masters Degree	12.8 %
	Doctoral Degree	0 %
Video Website Usage	Professional Degree	1.5 %
	Several times a day	60.9 %
	Once a day	14.3 %
	Several times a week	18 %
	Several times a month	4.5 %
	Less often	1.5 %
	Never	0.75 %

Table A.2.: Crowdsourcing Campaign C1 demographics (Part 2)

## A.2. File Listing

Next, we give a summary of the data attached to the thesis on the data medium. This includes on the one hand the source code of the implemented application and evaluation scripts, on the other hand the framework and video sequences used for the subjective evaluation.

Thesis File Listing	
Folder	Description
sources/cpp/dasvch	Implemented DASH client
sources/cpp/dasvch <sub>monitor</sub>	DASH Client GUI (optional, not required)
sources/evaluation/scripts	Scripts used in the testbed environment
sources/evaluation/traffic <sub>patterns</sub>	Utilized traffic patterns
sources/evaluation/matlab	Matlab evaluation scripts
sources/thesis	Latex source code of this thesis
userstudies/C1	Campaign 1 patterns
userstudies/C2	Campaign 2 patterns
userstudies/C3	Campaign 3 patterns
userstudies/C4	Campaign 4 patterns
userstudies/C5	Campaign 5 patterns

Table A.3.: Thesis Appendix File Listing



Campaign	Reward	#User	Users filtered
C1	0.30 \$	139	6
C2	0.30 \$	149	16
C3	0.30 \$	98	14
C4	0.30 \$	98	8
C5	0.30 \$	97	4

Table A.4.: Crowdsourcing campaigns number of users and money compensation

### A.3. Campaigns

Next, we present details about the conducted crowdsourcing campaigns. Table A.4 shows the number of user’s who participated, the number of users filtered due to the content questions and the monetary compensation for the campaigns 1 to 5.

Table A.5 gives an overview over the used switching patterns in the campaigns. The quality levels high and low are relative to the quality levels used for the specified amplitude. Figure A.1, A.2, A.3, A.4 and A.5 illustrate the utilized patterns.

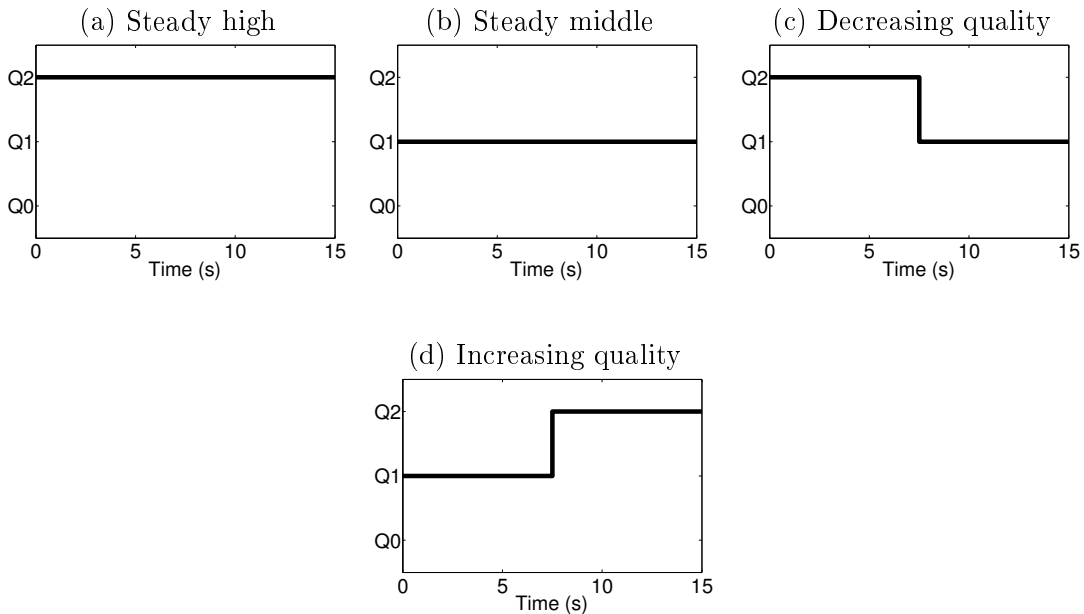


Figure A.1.: Campaign 1: Patterns *a* to *d*

### A.4. Web-based Crowdsourcing Interface

Next, we present the web-based crowdsourcing interface by example. Figure A.6 gives a screenshot of the page where the video sequence is shown and the user is

Campaign	Pattern	Switches	Amplitude	Begin	End	Scene(s)
C1	a	0	1	high	high	L,H
	b	0	1	low	low	L,H
	c	1	1	high	low	L,H
	d	1	1	low	high	L,H
	e	2	1	high	high	L,H
	f	2	1	low	low	L,H
C2	a	0	1	high	high	L
	b	1	1	high	low	L
	c	2	1	high	high	L
	d	3	1	high	low	L
	e	4	1	high	high	L
	f	6	1	high	high	L
	g	8	1	high	high	L
C3	a	0	2	high	high	L
	b	1	2	high	low	L
	c	2	2	high	high	L
	d	3	2	high	low	L
	e	4	2	high	high	L
	f	8	2	high	high	L
	g	14	2	high	high	L
	h	0	2	low	low	L
C4	a	0	2	high	high	L
	b	1	2	low	high	L
	c	2	2	low	low	L
	d	3	2	low	high	L
	e	4	2	low	low	L
	f	5	2	low	high	L
	g	7	2	low	high	L
	h	8	2	low	low	L
	i	0	2	low	low	L
C5	a	0	2	high	high	L
	b	1	2	high	low	L
	c	2	2	high	high	L
	d	3	2	high	low	L
	e	4	2	high	high	L
	f	5	2	high	low	L
	g	6	2	high	high	L
	h	8	2	high	high	L
	i	0	2	low	low	L

Table A.5.: Crowd-sourcing campaigns

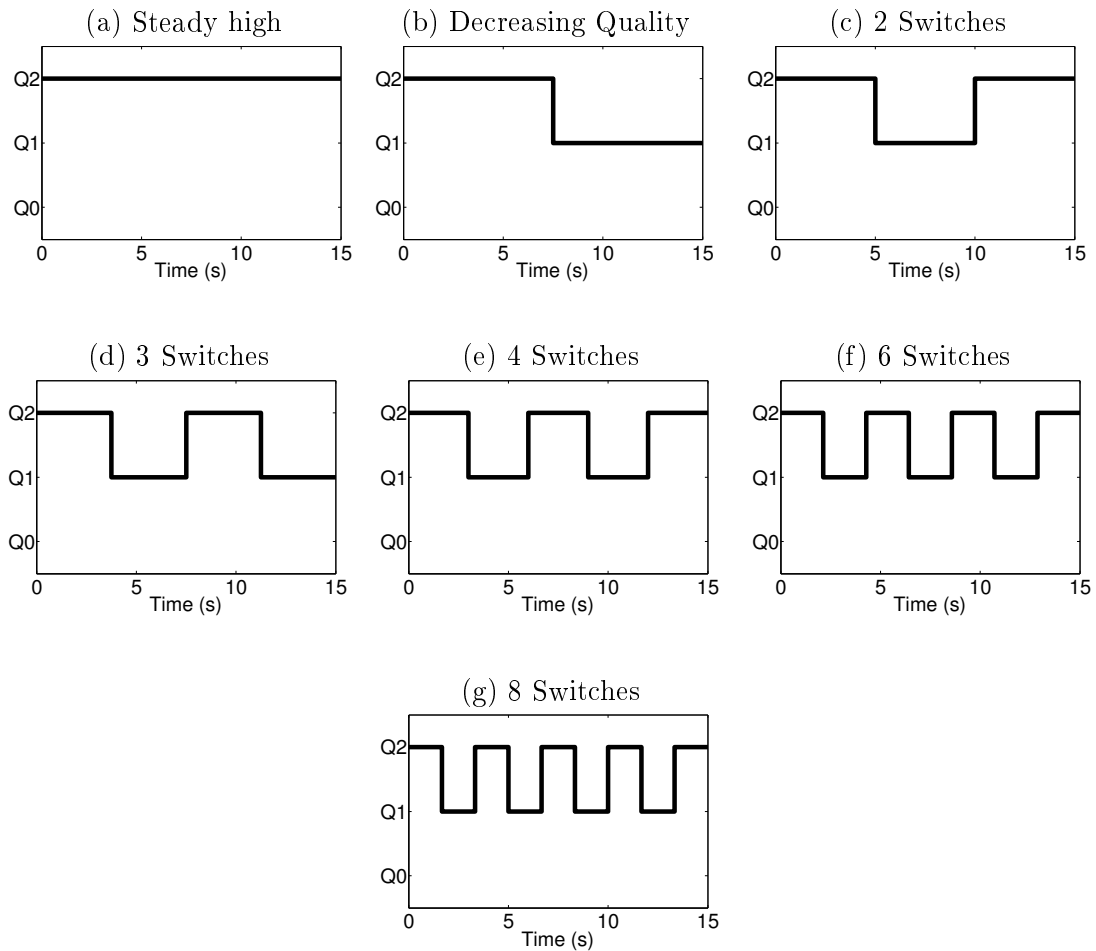


Figure A.2.: Campaign 2: Patterns *a* to *g*

asked to rate his viewing experience. The video sequence is shown in a resolution of 640 width to 480 height in the middle of the screen. The controls to start the playback and the indicating bar for the current playback position and the pre-buffer status is located underneath the video sequence. Below the playback controls, we show the rating slider to the user. After the video sequence finished the playback and the user rated his viewing experience, the user is able to press the next button to continue to the other questions (e.g. acceptance question).

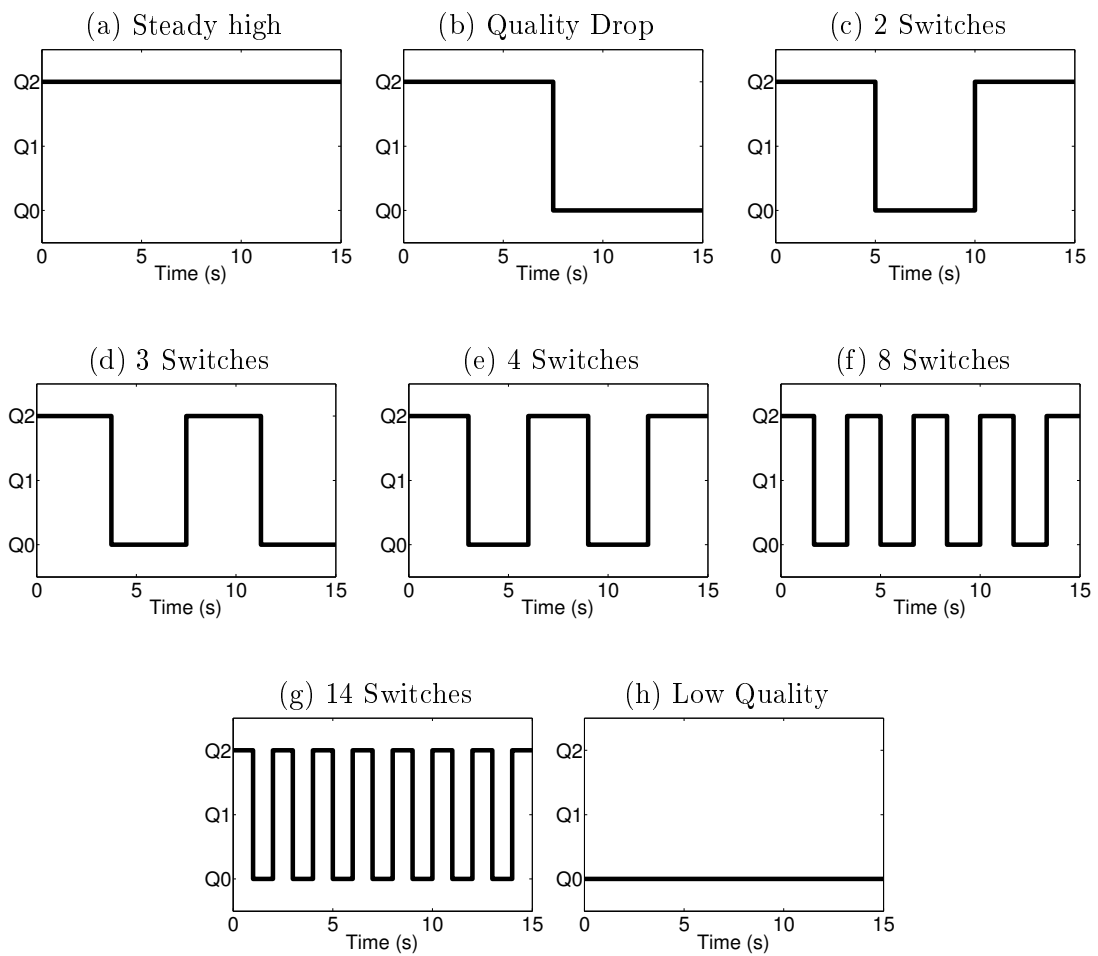


Figure A.3.: Campaign 3: Patterns *a* to *h*

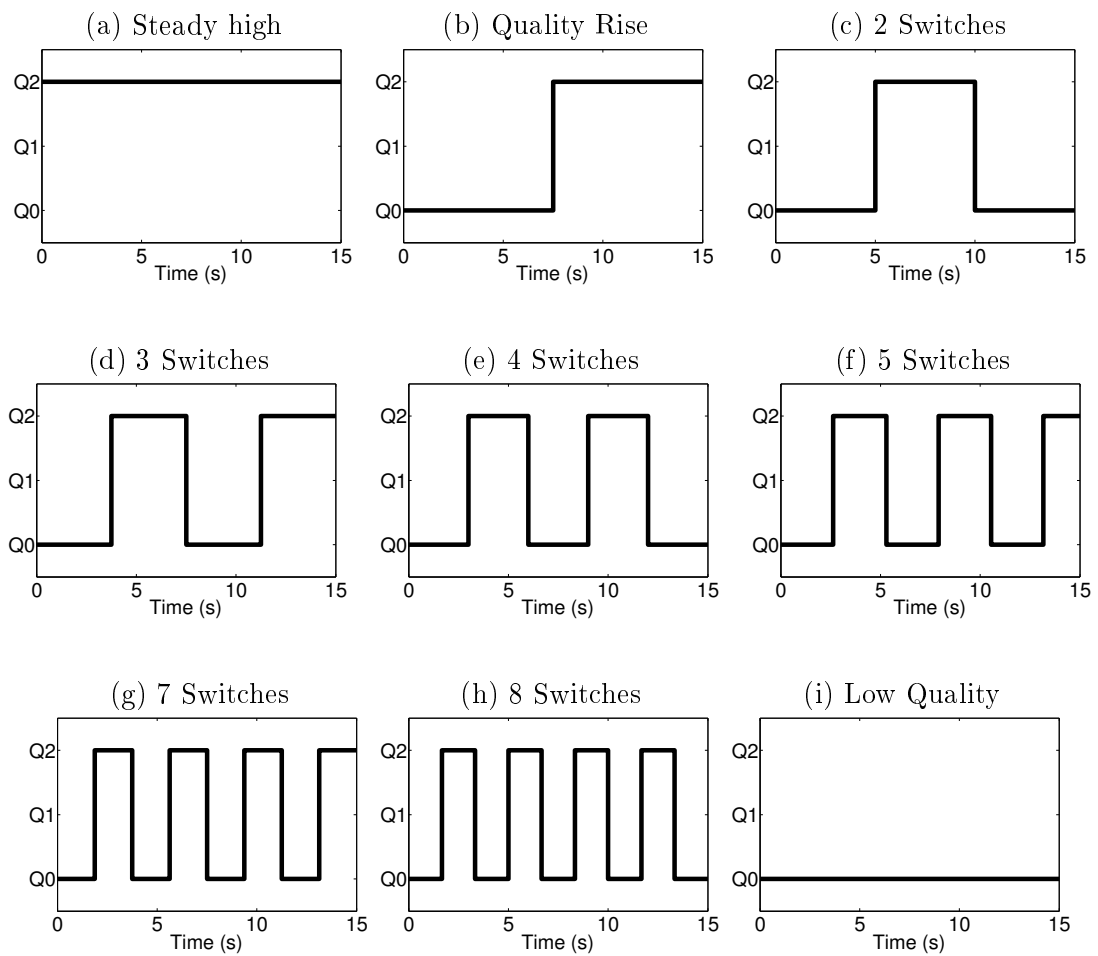


Figure A.4.: Campaign 4: Patterns  $a$  to  $i$

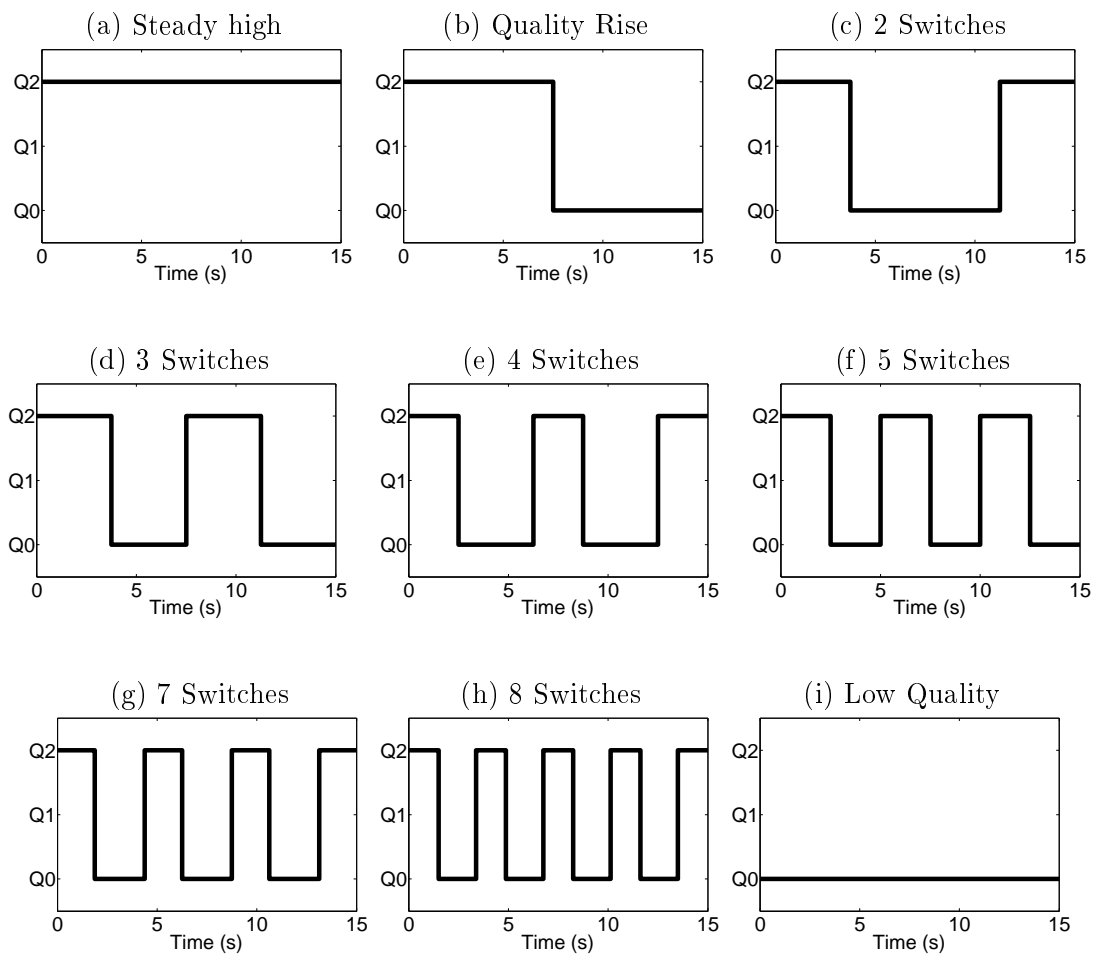



Figure A.5.: Campaign 5: Patterns  $a$  to  $i$

Step 5 of 34 **PREVIEW MODE, all data will be deleted**

## Viewing Experience



How would you rate the overall viewing experience?

Excellent  
Good  
Fair  
Poor  
Bad

Figure A.6.: Web-based Crowdsourcing Interface

## A.5. Crowdsourcing Questionnaire

Crowdsourcing Questionnaire (Part 1)	
Are you Male or Female?	Male Female
What is your age?	18 - 21 22 - 25 26 - 30 31 - 40 41 - 50 51 - 60 61+
What is the highest level of education you have completed?	Less Than High School High School/GED Some College 2-Year College Degree (Associates) 4-Year College Degree (BA, BS) Master's Degree Doctoral Degree Professional Degree (MD, JD) Not Listed
What is your current occupation?	Working Unemployed Student Apprenticeship Pensioner Home-keeper Not Listed
Are you wearing prescription glasses or contact lenses?	Yes No
On average, how long do you use the Internet per day?	Less than 1 hour/day 1 - 5 hours/day 5 - 7 hours/day 7 - 10 hours/day more than 10 hours/day
What is your main reason for using the Internet?	Professional (at work) For fun at home
Are you currently using a fixed or mobile Internet connection?	Fixed access line Mobile
Which continent do you live on?	Africa Asia Australia Europe North American South America



Crowdsourcing Questionnaire (Part 2)	
How often have you watched videos on web-sites like YouTube or Netflix during the last month?	Several times a day Once a day Several times a week Several times a month Less often Never

## A.6. Microworkers.com Campaign Description

In this task you will participate in a research survey about video quality. To participate your Internet download speed should be at least 1 Mbps (125 Kilobytes/s). The tasks takes quite long (up to 15 minutes), but we guarantee that everyone who finishes the task will be able to submit it at Microworkers.

1. Go to [http://132.187.12.59/Q1/{{MW\\_ID}}](http://132.187.12.59/Q1/{{MW_ID}})
2. Complete the survey
3. Watch the videos carefully and answer the questions
4. Submit you payment token here

## A.7. Crowdsourcing Campaign Introduction

The listing below gives the text displayed to the participants on the landing page of the online questionnaire.

Video Quality Assessment

Welcome to the video quality assessment of the Department of Communication Systems at the University of Wuerzburg, Germany. The survey will require you to watch and rate movie sequences streamed from the Internet. Your Internet download speed should be at least 1 Mbps (125 Kilobytes/s) to be able to participate in the survey. Adobe's FlashPlayer has to be installed to do the survey.

We will first ask you a few demographic questions before we will explain the survey procedure. Click next to continue.

After the landing page, the user is presented with the demographic questionnaire which is followed by the subsequent text.

How does it work?

Thank you for participating in the demographic survey!

The following survey is comprised of a sequence of 8 short movie clips in random order. Each clip is pre-loaded to avoid any unintended stallings during the playback. The pre-loading is indicated by a green/white bar.

After the pre-loading phase has finished a play button will replace the pre-loading bar. Click the play button to watch the video clip. After each clip you will be asked to answer 5 questions about the visual movie quality and sometimes also a question about the content. Please watch the movie clips carefully.

After the aforementioned text, the user is presented with three steps illustrating the procedure of the test. Step 1 (shown in Figure A.7) describes the pre-loading phase where a green bar is shown and the user has to wait for the pre-loading to finish. Step 2 (Figure A.8a) illustrates how to start the playback of the video sequence. Step 3 (Figure A.8b) describes how to use the rating slider.

### 1. Step

Wait for the green/white pre-loading bar to disappear. The pre-loading phase also includes the gray bar.



Figure A.7.: Crowdsourcing campaign introduction (Part 1)

### 2. Step:

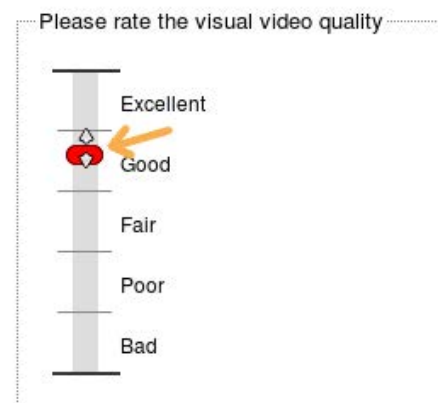
Press the play button underneath the video sequence and watch the video carefully.



(a) Crowdsourcing campaign introduction (Part 2)

### 3. Step:

Drag to slider to rate to answer the question.



Click next to continue to the video quality survey.

(b) Crowdsourcing campaign introduction (Part 3)

Figure A.8.: Low and high SITI scenes used in the user studies

# List of Figures

2.1. Example AVC GOP structure . . . . .	6
2.2. Possible Scalability Options for SVC ([66]) . . . . .	7
2.3. Media Description File (MPD) . . . . .	9
2.4. Tribler Download Strategy . . . . .	12
2.5. SVC scalability options and acceptable QoE [66] . . . . .	16
2.6. Example excerpt from an ACR test session . . . . .	18
3.1. Subjective studies overview with different research questions examined through different crowdsourcing campaigns (C1 - C5). . . . .	21
3.2. Perception of quality switches with different SITI values and amplitudes	25
3.3. Low and high SITI scenes used in the user studies . . . . .	28
3.4. Different quality levels used in QoE evaluation . . . . .	29
3.5. Number of users filtered by content question . . . . .	30
3.6. Actual Quality Switches and User Guesses . . . . .	32
3.7. Switches vs. Quality Rating for Campaigns 2 - 5 . . . . .	33
3.8. Time On High vs. Quality Rating for Campaigns 2 - 5 . . . . .	34
3.9. Switches vs. Acceptance Rate for Campaigns 2 - 5 . . . . .	34
3.10. Time On High vs. Acceptance Rate for Campaigns 2 - 5 . . . . .	35
3.11. Main effects plot for the quality rating . . . . .	36
4.1. Objective evaluation methodology overview . . . . .	38
4.2. Tears Of Steel representation bit-rates relative to the base layer . . .	39
4.3. Example buffer levels . . . . .	40
4.4. Example Switch from Q0 to Q1 . . . . .	41
4.5. Tears Of Steel with spatial scalability . . . . .	45
4.6. Testbed schematic . . . . .	46
4.7. Vehicular mobility pattern . . . . .	47
4.8. Evaluation Scenarios Summary . . . . .	49
5.1. Playback quality in time spend on the different quality levels . . . . .	52
5.2. Switching frequency . . . . .	53
5.3. Switching CDF . . . . .	53
5.4. Bandwidth utilization relative to a theoretical maximum . . . . .	54
5.5. Bandwidth wasted in percentage of movie file size on highest quality .	54
5.6. Memory usage . . . . .	55
5.7. Average playback quality . . . . .	56
5.8. Average switching frequency . . . . .	57
5.9. Difference between two concurrent clients . . . . .	58
5.10. Difference in playback quality for competing download traffic . . . . .	59

A.1. Campaign 1: Patterns <i>a</i> to <i>d</i> . . . . .	67
A.2. Campaign 2: Patterns <i>a</i> to <i>g</i> . . . . .	69
A.3. Campaign 3: Patterns <i>a</i> to <i>h</i> . . . . .	70
A.4. Campaign 4: Patterns <i>a</i> to <i>i</i> . . . . .	71
A.5. Campaign 5: Patterns <i>a</i> to <i>i</i> . . . . .	72
A.6. Web-based Crowdsourcing Interface . . . . .	73
A.7. Crowdsourcing campaign introduction (Part 1) . . . . .	76
A.8. Low and high SITI scenes used in the user studies . . . . .	76

# List of Tables

2.1. Comparison of DASH/HTTP and RTP/UDP . . . . .	10
2.2. Five-point rating scale . . . . .	19
3.1. Research questions implemented per campaign . . . . .	23
3.2. Pilot Study: Segmented Tears Of Steel Movie . . . . .	24
3.3. Crowd-sourcing campaigns . . . . .	30
4.1. Tears Of Steel with spatial scalability . . . . .	44
5.1. Comparison of the investigated algorithms . . . . .	61
A.1. Crowdsourcing Campaign C1 demographics (Part 1) . . . . .	65
A.2. Crowdsourcing Campaign C1 demographics (Part 2) . . . . .	66
A.3. Thesis Appendix File Listing . . . . .	66
A.4. Crowdsourcing campaigns number of users and money compensation	67
A.5. Crowd-sourcing campaigns . . . . .	68

# Bibliography

- [1] Amazon Mechanical Turk. <https://www.mturk.com/mturk/welcome>.
- [2] Blender Foundation. <http://www.blender.org/blenderorg/blender-foundation/>.
- [3] Boost C++ Libraries. <http://www.boost.org/>.
- [4] Cisco Visual Networking Index: Forecast and Methodology, 2012 - 2017. [http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-481360\\_ns827\\_Networking\\_Solutions\\_White\\_Paper.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html).
- [5] Dynamic adaptive streaming over HTTP (DASH), ISO/IEC 23009-1:2012. [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=57623](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57623).
- [6] Google Spdy Whitepaper.
- [7] ISO/IEC 14496-10:2012 - Part 10: Advanced Video Coding. [http://www.iso.org/iso/home/store/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=61490](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=61490).
- [8] ITU Telecommunication Standardization Sector (ITU-T). <http://www.itu.int/en/ITU-T/Pages/default.aspx>.
- [9] Joint Video Team. <http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jvt.aspx>.
- [10] libcurl. <http://curl.haxx.se/libcurl/>.
- [11] Linux Advanced Routing & Traffic Control. <http://lartc.org/>.
- [12] Microworkers.com. <http://microworkers.com/index.php>.
- [13] Moving Picture Experts Group (MPEG). <http://mpeg.chiariglione.org/>.
- [14] Peak signal-to-noise ratio. [http://en.wikipedia.org/w/index.php?title=Peak\\_signal-to-noise\\_ratio&oldid=540665991](http://en.wikipedia.org/w/index.php?title=Peak_signal-to-noise_ratio&oldid=540665991).
- [15] pugixml. <http://code.google.com/p/pugixml/>.
- [16] Wikipedia: Creative Commons License. [http://en.wikipedia.org/w/index.php?title=Creative\\_Commons\\_license&oldid=542607663](http://en.wikipedia.org/w/index.php?title=Creative_Commons_license&oldid=542607663).

- [17] Wikipedia: Sobel. [http://en.wikipedia.org/w/index.php?title=Sobel\\_operator&oldid=549725924](http://en.wikipedia.org/w/index.php?title=Sobel_operator&oldid=549725924).
- [18] Wikipedia: SPDY. <http://en.wikipedia.org/w/index.php?title=SPDY&oldid=546929833>.
- [19] ITU-T Video Coding Experts Group, ISO/IEC Moving Pictures Experts Group, Joint Video Team. JSVM Reference Software. <http://www.hhi.fraunhofer.de/de/kompetenzfelder/image-processing/research-groups/image-video-coding/svc-extension-of-h264avc/jsvm-reference-software.html>.
- [20] Adobe Systems Inc. Adobe HTTP Dynamic Streaming. <http://www.adobe.com/products/hds-dynamic-streaming.html>.
- [21] D. Alfonso, B. Biffi, and L. Pezzoni. Adaptive gop size control in h.264/avc encoding based on scene change detection. In *Proceedings of the 7th Nordic Signal Processing Symposium, 2006. NORSIG 2006*, 2006.
- [22] Apple Inc. HTTP Live Streaming. <https://developer.apple.com/resources/http-streaming/>.
- [23] Blender Institute. Tears Of Steel. <http://www.tearsofsteel.org/>.
- [24] ITU-T RECOMMENDATION BT.500-11. Methodology for the subjective assessment of the quality of television pictures.
- [25] Shelley Buchinger, Simone Kriglstein, and Helmut Hlavacs. A comprehensive view on user studies: survey and open issues for mobile tv. In *Proceedings of the seventh european conference on European interactive television conference*, 2009.
- [26] Ming-Jun Chen and Alan C Bovik. Fast structural similarity index algorithm. *Journal of Real-Time Image Processing*, 6(4), 2011.
- [27] Florin Dobrian, Asad Awan, Dilip Joseph, Aditya Ganjam, Jibin Zhan, Vyas Sekar, Ion Stoica, and Hui Zhang. Understanding the impact of video quality on user engagement. *SIGCOMM-Computer Communication Review*, 41(4), 2011.
- [28] Rushabh Doshi and Pei Cao. Streaming traffic fairness over low bandwidth wan links. In *Proceedings of The Third IEEE Workshop on Internet Applications. WIAPP 2003*. IEEE, 2003.
- [29] Alexander Eichhorn and Pengpeng Ni. Pick your layers wisely-a quality assessment of h. 264 scalable video coding for mobile devices. In *IEEE International Conference on Communications*, 2009.
- [30] Randa El-Marakby and David Hutchison. A scalability scheme for the real-time control protocol. In *High Performance Networking*, 1998.

- [31] Rosario Feghali, Demin Wang, Filippo Speranza, and André Vincent. Quality metric for video sequences with temporal scalability. In *IEEE International Conference on Image Processing, 2005. ICIP 2005*, volume 3, 2005.
- [32] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: a new tcp-friendly high-speed tcp variant. *ACM SIGOPS Operating Systems Review*, 42(5), 2008.
- [33] Selig Hecht. The visual discrimination of intensity and the weber-fechner law. *The Journal of General Physiology*, 7(2), 1924.
- [34] Tobias Hoßfeld, Sebastian Egger, Raimund Schatz, Markus Fiedler, Kathrin Masuch, and Charlott Lorentzen. Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea. In *QoMEX 2012*, Yarra Valley, Australia, July 2012.
- [35] Tobias Hoßfeld, Christian Keimel, Matthias Hirth, Bruno Gardlo, Julian Habigt, Klaus Diepold, and Phuoc Tran-Gia. CrowdTesting: A Novel Methodology for Subjective User Studies and QoE Evaluation. Technical report, University of Würzburg, February 2013.
- [36] Tobias Hoßfeld, Raimund Schatz, Michael Seufert, Matthias Hirth, Thomas Zinner, and Phuoc Tran-Gia. Quantification of YouTube QoE via Crowdsourcing. In *IEEE International Workshop on Multimedia Quality of Experience - Modeling, Evaluation, and Directions (MQoE 2011)*, Dana Point, CA, USA, December 2011.
- [37] Tobias Hoßfeld, Dominik Strohmeier, Alexander Raake, and Raimund Schatz. Pippi Longstocking Calculus for Temporal Stimuli Pattern on YouTube QoE. In *5th ACM Workshop on Mobile Video (MoVid 2013)*, Oslo, Norway, February 2013.
- [38] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13), 2008.
- [39] ITU-T. Rec. P.800: Methods for subjective determination of transmission quality.
- [40] P.910 ITU-T RECOMMENDATION. Subjective video quality assessment methods for multimedia applications.
- [41] Sung Ho Jin, Cheon Seog Kim, Dong Jun Seo, and Yong Man Ro. Quality measurement modeling on scalable video applications. In *MMSP 2007. IEEE 9th Workshop on Multimedia Signal Processing*, 2007.
- [42] Christian Keimel, Julian Habigt, Clemens Horch, and Klaus Diepold. Qualitycrowd - a framework for crowd-based quality evaluation. In *Picture Coding Symposium 2012 (PCS2012)*, May 2012.



- [43] Jong-Seok Lee, Francesca De Simone, Naeem Ramzan, Zhijie Zhao, Engin Kurutepe, Thomas Sikora, Jörn Ostermann, Ebroul Izquierdo, and Touradj Ebrahimi. Subjective evaluation of scalable video coding for content distribution. In *Proceedings of the international conference on Multimedia*. ACM, 2010.
- [44] John D McCarthy, M Angela Sasse, and Dimitrios Miras. Sharp or smooth?: comparing the effects of quantization vs. frame rate for streamed video. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2004.
- [45] Microsoft Corporation. Smooth Streaming. <http://www.iis.net/downloads/microsoft/smooth-streaming>.
- [46] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz. Adaptation algorithm for adaptive streaming over http. In *19th International Packet Video Workshop (PV 2012)*, Munich, Germany, may 2012.
- [47] C. Müller, S. Lederer, and C. Timmerer. An evaluation of dynamic adaptive streaming over http in vehicular environments. In *4th Workshop on Mobile Video (MoVID 2012)*, Chapel Hill / USA, February 2012.
- [48] P. Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen. Flicker effects in adaptive video streaming to handheld devices. In *19th ACM int. conf. on Multimedia (MM 11)*, Scottsdale, AZ, USA, November 2011.
- [49] Pengpeng Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen. Spatial flicker effect in video scaling. In *2011 Third International Workshop on Quality of Multimedia Experience (QoMEX)*, sept. 2011.
- [50] Simon Oechsner, Thomas Zinner, Jochen Prokopetz, and Tobias Hofffeld. Supporting scalable video codecs in a P2P video-on-demand streaming system. In *21th ITC Specialist Seminar on Multimedia Applications - Traffic, Performance and QoE*, Miyazaki, Japan, March 2010.
- [51] Sebastian Möller Patrick Le Callet and eds. Andrew Perkis. Qualinet White Paper on Definitions of Quality of Experience (Version 1.2). In *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, March 2013.
- [52] Margaret H Pinson and Stephen Wolf. Comparing subjective video quality testing methodologies. In *Visual Communications and Image Processing 2003*. International Society for Optics and Photonics, 2003.
- [53] David C Robinson, Yves Jutras, and Viorel Craciun. Subjective video quality assessment of http adaptive streaming technologies. *Bell Labs Technical Journal*, 16(4), 2012.
- [54] J. Rosenberg and Henning Schulzrinne. Timer reconsideration for enhanced rtp scalability. In *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, 1998.

- [55] Yago Sánchez de la Fuente, Thomas Schierl, Cornelius Hellge, Thomas Wiegand, Dohy Hong, Danny De Vleeschauwer, Werner Van Leekwijck, and Yannick Le Louédec. idash: improved dynamic adaptive streaming over http using scalable video coding. In *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011.
- [56] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. Overview of the scalable video coding extension of the h.264/avc standard. 2007.
- [57] Heiko Schwarz and Thomas Wiegand. Rd optimized multi-layer encoder control for svc. In *IEEE International Conference on Image Processing, 2007. ICIP 2007*, volume 2. IEEE, 2007.
- [58] C Andrew Segall and Gary J Sullivan. Spatial scalability within the h. 264/avc scalable video coding extension. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9), 2007.
- [59] Kalpana Seshadrinathan, Rajiv Soundararajan, Alan Conrad Bovik, and Lawrence K Cormack. Study of subjective and objective quality assessment of video. *Image Processing, IEEE Transactions on*, 19(6), 2010.
- [60] Zhou Wang, Alan C Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004.
- [61] Zhou Wang, Ligang Lu, and Alan C Bovik. Video quality assessment based on structural distortion measurement. *Signal processing: Image communication*, 19(2), 2004.
- [62] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, 2003*, volume 2, 2003.
- [63] Mathias Wien, Heiko Schwarz, and Tobias Oelbaum. Performance analysis of svc. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9), 2007.
- [64] Feng Xiao et al. Dct-based video quality evaluation. *Final Project for EE392J*, 2000.
- [65] Michael Zink, Oliver Künzel, Jens Schmitt, and Ralf Steinmetz. Subjective impression of variations in layer encoded videos. In *11th Int. Conf. on Quality of service, IWQoS'03*, Berkeley, CA, USA, 2003.
- [66] Thomas Zinner. Performance modeling of qoe-aware multipath video transmission in the future internet. 2012.