University of Würzburg
Institute of Computer Science
Research Report Series

Transport Protocol Influences on YouTube Videostreaming QoE

Tobias Hoßfeld[1,2], Raimund Schatz[2], Thomas Zinner[1],
Michael Seufert[1], Phuoc Tran-Gia[1]

Report No. 482 September 2011

[1] University of Würzburg, Institute of Computer Science, Chair of Communication Networks
Am Hubland, 97074 Würzburg, Germany
hossfeld@informatik.uni-wuerzburg.de

[2] Telecommunications Research Center Vienna - FTW
A-1220 Vienna, Austria
schatz@ftw.at

Acknowledgments. This work was conducted within the Internet Research Center (IRC) at the University of Würzburg. The work has been supported by COST TMA Action IC0703, COST IC1003 European Network on Quality of Experience in Multimedia Systems and Services (QUALINET)(http://www.qualinet.eu/), the project G-Lab, funded by the German Ministry of Educations and Research (Förderkennzeichen 01 BK 0800, G-Lab), and the project ACE 2.0 funded within the program COMET - Competence Centers for Excellent Technologies by BMVIT, BMWA, and the City of Vienna. The COMET program is managed by the FFG. The authors alone are responsible for the content of the paper.

Transport Protocol Influences on YouTube Videostreaming QoE

Tobias Hoßfeld[1,2], Raimund Schatz[2], Thomas Zinner[1], Michael Seufert[1], Phuoc Tran-Gia[1]

[1]University of Würzburg, Institute of Computer Science, Chair of Communication Networks
Am Hubland, 97074 Würzburg, Germany
hossfeld@informatik.uni-wuerzburg.de

[2]Telecommunications Research Center Vienna - FTW
A-1220 Vienna, Austria
schatz@ftw.at

Abstract

Video streaming currently dominates global Internet traffic and will be of even increasing importance in the future. In this paper we assess the impact of the underlying transport protocol on the user perceived quality for video streaming using YouTube as example. In particular, we investigate whether UDP or TCP fits better for Video-on-Demand delivery from the end user's perspective, when the video is transmitted over a bottleneck link. For UDP based streaming, the bottleneck link results in spatial and temporal video artifacts, decreasing the video quality. In contrast, in the case of TCP based streaming, the displayed content itself is not disturbed but playback suffers from stalling due to rebufferung.

Due to the lack of existing Quality of Experience (QoE) models for online video services that are based on TCP-streaming, we propose a generic subjective QoE assessment methodology for multimedia applications (like online video) that is based on crowdsourcing - a highly cost-efficient, fast and flexible way of conducting user experiments. We demonstrate how our approach successfully leverages the inherent strengths of crowdsourcing while addressing critical aspects such as the reliability of the experimental data obtained. As a result, we present a dedicated QoE model for YouTube that takes into account the key influence factors (such as stalling events caused by network bottlenecks) that shape quality perception of this service.

The results of subjective user studies for both scenarios (UDP based on related work, TCP based on own studies) are analyzed in order to assess the transport protocol influences on Quality of Experience of YouTube. To this end, application-level measurements are conducted for YouTube streaming over a network bottleneck in order to develop models for realistic stalling patterns. Furthermore, mapping functions are derived that accurately describe the relationship between network-level impairments and QoE for both protocols.

1 Introduction

Video streaming dominates global Internet traffic and is expected to account for 57 % of all consumer Internet traffic in 2014 generating over 23 exabytes per month [1]. It can be distinguished between delivery of live video streaming with on-the-fly encoding, like IPTV or Facetime, and delivery of pre-encoded video, so called Video-on-Demand (VoD). The most prominent video streaming portal is Youtube which serves more than

two billion videos daily [2]. YouTube videos are delivered over the Internet by means of the HTTP protocol which is actually used for the majority of the residential broadband Internet traffic [3] as illustrated in Figure 1.

However in practice, many users face volatile performance of the service, e.g. due to bad network conditions or congested media streaming servers. Such adverse conditions are the main causes for bad online video Quality of Experience (QoE). Network and service providers need to be able to observe and react upon quality problems, at best before the customer takes notice of them. Therefore, appropriate QoE models and metrics are required, like those provided by this work on YouTube video streaming.

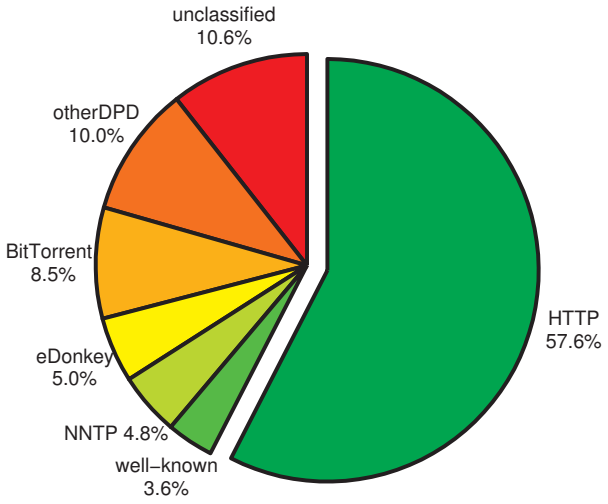

Figure 1: Application mix for Internet traffic [3]

The transport of video streams in the Internet is currently realized either with TCP or UDP. However, due to the diverse features of these protocols their application has a huge impact on the streaming behavior. In the domain of video streaming, traditional UDP-based services like IPTV or Real Media streaming typically do not guarantee packet delivery. Thus, congestion in the network or at the multimedia servers leads to lost packets causing visual artifacts, jerky motion or jumps in the stream, forms of degraded media quality which have been extensively studied in previous video quality research. In contrast, delivery of YouTube video to the end user is realized as progressive download using TCP as transport protocol. The usage of TCP guarantees the delivery of unaltered video content since the protocol itself cares for the retransmissions of corrupted or lost packets. Further, it adapts the transport rate to network congestion, effectively minimizing packet loss. However, if available bandwidth is lower than the video bit rate, video transmission becomes too slow, gradually emptying the playback buffer until underrun occurs. If rebuffering happens, the user notices interrupted video playback, commonly referred to as *stalling*. In this respect, YouTube QoE is different from traditional UDP-based video streaming, since with TCP only the video playback itself is disturbed while the transmitted audiovisual content remains unaltered.

The question arises which transport protocol is more appropriate from the end user's point of view, i.e. the Quality of Experience. To answer this question we consider a bottleneck scenario in which network capacity is limited. Thus, the available network bandwidth may be lower than the required video bit rate and the user may suffer from stalling and quality degradation for TCP and UDP, respectively. In order to compare the impact of the transport protocols on the QoE, two subjective user studies are presented which allow quantifying the impact of the bottleneck link capacity on QoE. [4] executed user surveys to evaluate QoE of video streaming with lost packets. However, due to the current lack of QoE models that identify key influence factors for YouTube (e.g. demographics of users, Internet application usage habits, content types, network impairments) and explicitly address stalling effects in the context of TCP-based online video, own subjective user studies need to be performed. For deriving a QoE model, crowdsourcing therefore seems to be an appropriate alternative approach. With crowdsourcing, subjective user studies can be efficiently conducted at low costs with adequate user numbers for obtaining statistically significant QoE scores [5]. However, reliability of results cannot be assumed because of the anonymity and remoteness of participants (cf. [6] and references therein): some subjects may submit incorrect results in order to maximize their income by completing as many tasks as possible; others just may not work correctly due to lack of supervision. Therefore, it is necessary to develop an appropriate methodology that addresses these issues and ensures consistent behavior of the test subjects throughout a test session and thus obtain reliable QoE results.

The contribution of this technical report has different facets. (1) An intensive YouTube measurement study is conducted in order to quantify the relevant application-level QoS parameters for YouTube over a bottleneck. In particular, the observed stalling patterns are modeled in terms of stalling frequency and stalling length. (2) Then, we provide a YouTube QoE model taking into account stalling as key influence factor based on subjective user studies. (3) To this end, we develop a generic subjective QoE testing methodology for Internet applications like YouTube based on crowdsourcing for efficiently obtaining highly valid and reliable results. (4) Finally, YouTube video streaming via TCP and via UDP is compared from the end-user perspective by means of subjective user studies (cf. Section 3 and [4]). The comparison is realized by transforming the results of the subjective tests to the common denominator in the considered scenario, that is the network bandwidth limitation due to the bottleneck. Since we provide first a YouTube QoE model for realistic stalling pattern, the work presented here is the first comparing QoE – and in particular YouTube QoE – for different transport protocols.

The remainder of this paper is structured as follows. Section 2 shows the application-level measurements for YouTube over a bottleneck. This includes the video characteristics in terms of duration and video bit rate as well as the observed stalling patterns which is required to later map the bottleneck bandwidth to QoE. The subjective user study on QoE for YouTube video streaming in the presence of stalling, which means via TCP, is reviewed in Section 3. Section 3.1 gives a background on crowdsourcing and the platform used in this work. The subjective test methodology is presented in Section 3.2 aiming at an appropriate test design to detect unreliable user ratings. In Section 3.3, the test results are statistically analyzed. In particular, we apply different results filtering

levels and assess the reliability of the data set. The YouTube QoE is then quantified in Section 3.4 for a realistic impairment scenario, where the YouTube video is streamed over a bottleneck link. The QoE model for UDP based transmission of YouTube videos is presented in Section 4 which also comapres the results of the subjective tests. Finally, Section 5 concludes this work and discusses further research issues.

2 Measurement of YouTube Stalling Patterns on Application Level

In the considered bottleneck scenario for TCP, the available network bandwidth B is limited. When downloading a video which is encoded at a video bit rate $V > B$, stalling may occur. The number N of stallings during the video playout as well as the length L of a single stalling event will both affect the QoE. However, the stalling pattern even in the bottleneck scenario with constant network capacity may be quite complex, since several factors interact and influence the observed stalling pattern, (a) YouTube's implementation of flow control on application layer [7], (b) TCP's flow control on transport layer, (c) variabe bit rate due to the used video encoding, (d) implementation of the video player and its video buffer.

Therefore, we derive in the following a simple model for the observed stalling patterns based on an application-level measurement study. In Section 2.1, the measurement setup is explained. Section 2.2 takes a closer look at the characteristics of YouTube videos in terms of video bit rate V and the duration D of the video clips. The observed stalling patterns over the dedicated bottleneck are analyzed in Section 2.3. The notation and variables frequently used throughout this paper are summarized in Table 1.

2.1 Setup of Application-Level Measurements

Our YouTube TCP measurement campaign took place from July to August, 2011 during which more than 37 000 YouTube videos were requested, about 35 GByte of data traffic was captured, and more than 1 000 videos were analyzed frame by frame in detail. In addition, 266 245 video descriptions were downloaded from YouTube containing the duration of the videos.

For measuring YouTube video streaming over a bottleneck, the measurement setup included three different components. (1) *Bandwidth shaper.* A network emulation software was used to limit the upload and download bandwidth. In our experiments, the "NetLimiter" bandwidth shaper was applied. (2) *YouTube user simulation.* This component simulated a user watching YouTube videos in his browser. Therefore, a local Apache web server was configured and web pages were dynamically generated, which call the YouTubi API for embedding and playing the YouTube video. The embedding of the YouTube videos in an own web page is necessary for monitoring the applictionlevel QoS. In order to obtain a random snapshot on YouTube, we randomly searched for videos via the YouTube API and used a public dictionary of english words as keyword for the YouTube search request. (3) *QoS monitor.* The video player status ("playing",

Variables	
V	total bit rate of video in (kbps)
D	duration of video in (s)
B	bandwidth limitation in (kbps)
N	number of stalling events
L	duration of a single stalling event
F	stalling frequency $F = {}^{N}/_{D}$ in (1/s)
R	packet loss ratio
ρ	throughput normalized by video bitrate, i.e. $\rho = {}^{B}/_{V}$
Functions	
$f_L(N)$	mapping function between number N of stalling events and MOS values for stalling events of length L via TCP
$g_v(R)$	mapping function between packet loss ratio R and MOS values for videos with resolution v (CIF, 4CIF) via UDP
$\Upsilon_L(\rho)$	mapping function between normalized throughput ρ and MOS values for stalling events of length L via TCP
$\Upsilon_v(\rho)$	mapping function between normalized throughput ρ and MOS values for videos with resolution v (CIF, 4CIF) via UDP
μ_X	average value of measurements X
σ_X	standard deviation of measurements X

Table 1: Notation and variables frequently used

"buffering", "ended") and the used buffer size (in terms of number of bytes loaded for the current video) were monitored within the generated web page using Javascript. At the end of the simulation (i.e. when the simulated user completely watched the video, after a certain timeout, or in case of any player errors), the stalling monitoring information and the buffer status were written to a logfile. In addition, the network packet traces were captured using wireshark and tshark. As a result, both network-level QoS parameters (from the packet traces) and application-level QoS parameters (the stalling patterns) were captured.

The QoS monitor component provided the data for analyzing the stalling pattern on application level. The YouTube API specifies an event called "onStateChange" which is fired whenever the state of the player changes. For each event, e.g. when the video player switches between buffering of data and playing the video, the current timestamp, the number of bytes loaded, as well as an identifier for the event itself are recorded by the QoS monitor. However, it has to be noted that the timer resolution depends on the actual JavaScript implementation within the used browser. In our experiments, we used the Internet explorer within Windows 7 which shows a timer resolution of about 16 ms.

For analyzing the video files, the video contents were extracted from the packet traces. The YouTube API specifies a set of calls for requesting videos via HTTP. Via pattern matching, these HTTP requests and corresponding HTTP objects were identified. Fur-

thermore, YouTube uses DNS translation and URL redirection, as the actual video contents are located on various caching servers, see [8, 9, 10]. The video contents were then reassembled from the corresponding TCP stream.

The video file itself was parsed by implementing a perl module which analyzed the video frames and extracted meta-information from the video file. As a result, video information like video bit rate, video resolution, used audio and video codecs, or video size and duration were extracted. Furthermore, for each video frame in the video stream, information about the video playback times of frames, the size of the video frames, as well as the type of frames (key frame or interframe) were extracted.

2.2 Video Characteristics

The characteristics of YouTube videos were analyzed in terms of bit rate V and duration D, which both influence the actual stalling pattern. There are already several works considering the YouTube video durations. [11] showed that 97.9 % of the videos are shorter than 10 min and 99.1 % are shorter than 700 s. The coefficient of variation of the video length was found to be about 1, while the mean duration was about 4.15 min, see [12]. In previous work [13], we measured a mean duration of 5.65 min and a coefficient of variation about 1,24. In 2010, however, YouTube increased the upload limit of the video duration to 15 min. Therefore, it is worth take a closer look on the impact of these changes.

The statistical analysis of the video durations D showed that 3.12 % of the videos were longer than 15 min. About 0.04 % of the videos were empty and had a length of 0 s. From the regular videos, i.e. shorter than 15 min, the average duration is 5.54 min and the coefficient of variation is about 1,65. Thus, the average duration is quite close, while the variance slighthly increased compared to our previous measurements in 2008 [13]. In addition, we found that the video duration can be well fitted by a Weibull distribution with parameters $a = 288.52$ and $b = 1.52$.

Next, the bit rates of the videos are analyzed. Figure 2 shows the cumulative distribution function (CDF) of the bit rate of the YouTube videos. Since a video typcially consists of an audio stream as well as a video stream, it is differentiated between the audio bit rate, the video bit rate and the total data rate. However, the audio stream typically only takes a fraction of the entire data rate which lies between 10 % and 20 %. The audio bit rate A also shows only small variances across the different YouTube videos with a standard deviation of $\sigma_A = 29.58$ kbps. In contrast, the video bit rate shows larger variances (209.95 kbps) and the video stream clearly determines the total data rate. The correlation between the total data rate and the video bit rate is about 0,99, while the audio data rate and total data rate is uncorrelated (with a Pearson linear correlation coefficient of 0,36). We also found that the different bit rates (audio, video, total) can be well approximated by a Weibull distribution. The corresponding CDFs are depicted as solid lines in Figure 2; the CDFs of the measured bit rates are plotted as dashed lines. The parameters a and b of the Weibull distribution as well as the mean and standard deviation of the measured data are given in Table 2.

In the following, we only consider the total data rate, since the video playout will stall,

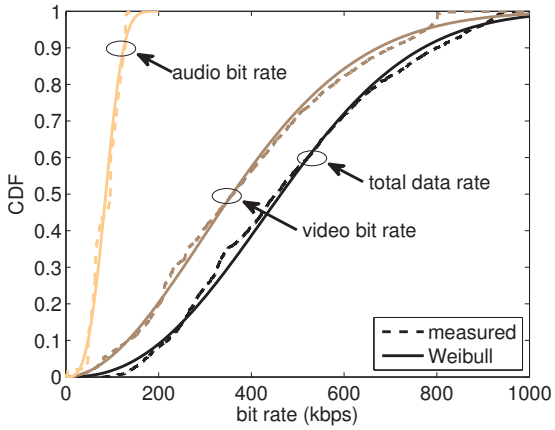

Figure 2: Measured audio, video, and total data rates as well as their corresponding Weibull distributions with parameters obtained by minimizing least square errors

type	*mean (kbps)*	*std. (kpbs)*	*a*	*b*
total	478.88	216.88	541.62	2.37
video	385.99	209.95	434.82	1.91
audio	86.73	29.58	96.30	3.22

Table 2: Measured audio and video bit rates and parameters a and b of the fitted Weibull distribution

if either the audio or the video data is not delivered on time. For the sake of readability, we will refer to the total data rate V as video bit rate in order to avoid confusions with the network data rate limit B.

2.3 Observed Stalling Patterns over Bottleneck

The aim of this section is to model the observed stalling patterns when the YouTube video is streamed over a bottleneck. The subjective user studies in Section 3 quantify QoE depending on the number N of stalling events and the length L of a single stalling event. Thus, a mapping function $f_L(N)$ between the stalling parameters as application-level QoS and the QoE in terms of mean opinion score (MOS) values is provided. Thus, we derive the influence of the bottleneck capacity B on the observed stalling pattern in the following. In particular, we depict two exemplary bandwidth limitations, that are $B = 384\,\text{kbps}$ as typical bandwidth of UMTS cell phones and $B = 450\,\text{kbps}$ which is roughly the median of the video bit rate V.

2.3.1 Influence of Video Bit Rates

Figure 3 shows again the CDF of the video bit rate, based on two experiments run at 384 kbps and 450 kbps. However, we also distinguish whether stalling occurs or not during the video playout. In the experiment with $B = 384$ kbps, 300 videos were completely downloaded and analyzed. No stalling occured for 116 videos corresponding to 38.67 % of all videos. In this case, the video bit rate is mostly smaller than bottleneck capacity, i.e. $V < B$. However, there were two videos without stalling, although the video bit rate was significantly larger than B. In that case, the video durations of these two videos were quite short with $D = 10.8$ s and $D = 9.8$ s, respectively. Since the video player has implemented a video playout buffer, sufficient data is downloaded before the video playout starts and no stalling occurs.

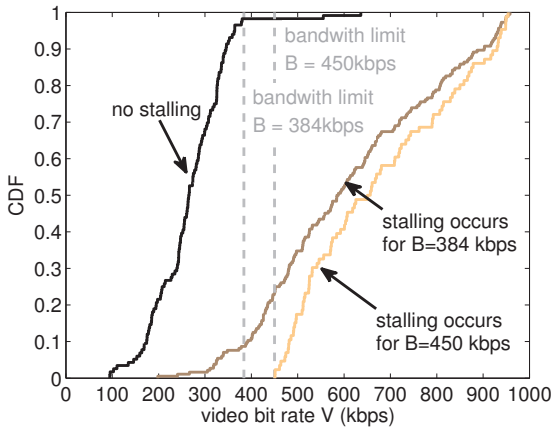

Figure 3: Stalling occurrence depending on video bit rate for two different bottleneck capacities

Figure 3 shows another interesting phenomenon. For some videos with video bit rate $V < B$, stalling still occurs although the network capacity is sufficient to download the entire video data during the playout time of the video. In that case, stalling is caused by the variability of the video bit rate. It has to be noted that in Figure 3 the results for the bandwidth limitation of $B = 450$ kbps do not show this feature, since the videos were already filtered according to their bit rate V so that $V > B$. This filtering was done in order to decrease number of experimental runs where no stalling, occurs as we are primarily interested in modeling the actual stalling patterns.

So far, we have investigated under which conditions stalling occurs or not. For quantifying the impact of stalling on QoE in case of TCP based video delivery, however, the stalling pattern (and correlations between different factors) has to be described statistically. In particular, the number N of stalling events per video clip as well as the duration L of the stalling events is of interest.

2.3.2 Stalling Frequency

Next, the stalling frequency F is analyzed which is defined as the ratio of the number of stalling events and the duration D of the video, i.e. $F = {}^{N}/_{D}$. First, the correlation of F with several influence factors was investigated. In particular, the following variables were considered with Pearson's linear correlation coefficient given in brackets: 1. frame rate (-0.03), 2. video duration (-0.35), 3. median of stalling length (0.37), 4. number of stallings (0.47), 5. mean stalling length (-0.58), 6. video bit rate (0.87). Thus, there is no significant correlation between stalling frequency and frame rate, number of stalling, the video duration or the stalling length. The stalling frequency is strongly correlated only with the video bit rate.

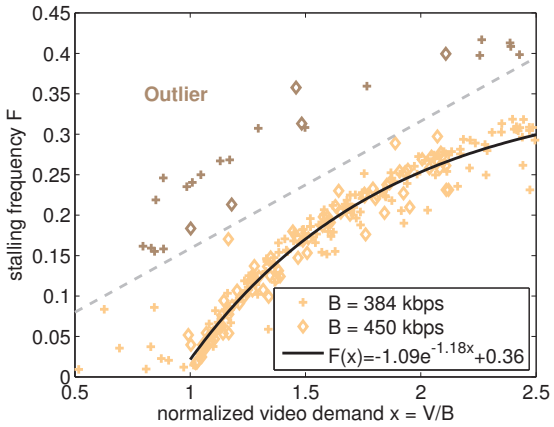

Figure 4: Measured and fitted stalling frequency F depending on the normalized video demand x as ratio of video bit rate V and bottleneck capacity B

Figure 4 depicts the stalling frequency depending on the normalized video demand x for two different bandwidth limitations. The normalized video demand is defined as the ratio of the video bit rate V and the bottleneck capacity B, i.e. $x = {}^{V}/_{B}$. The measurement results for each video clip are plotted with "$\diamond$" marker and "+" marker for $B = 384\,\text{kbps}$ and $B = 450\,\text{kbps}$, respectively. As a result, we see that the measurement results – for both bottleneck capacities – lie in the same area. In particular, the measured frequencies with the corresponding measured video demands can be well fitted by an exponential function which we found by minimizing the least square errors,

$$F(x) = -1.09e^{-1.18x} + 0.36\,. \tag{1}$$

The resulting coefficient of determination of the fitting function F and the measurement data is $D = 0.943$. However, there are several outliers which lie above the dashed line in Figure 4. About 15.22 % of the video clips are assumed to be outliers. We found no statistical correlation between these values of F and any other variables. An in-depth analysis of the packet traces as well as of the video contents did not reveal a clear reason for this. However, we assume that these outliers are caused by the implementation of

the video player itself. Considering the correlation coefficients of F and the video bitrate V without the outliers leads to 0,955 and 0,958 for $B = 384\,\text{kbps}$ and $B = 450\,\text{kbps}$, respectively.

Thus, when the bottleneck capacity is equal to the video bit rate, i.e. $x = 1$, the stalling frequency is $F(1) = 0.021$. In that case, a one minute video clip will already stall once due to the variable video bit rate, see Section 2.3.1. According to the curve fitting function, the stalling frequency will converge and it is $\lim_{x\to\infty} F(x) = 0.357$. Hence, a one minute clip will stall at most 21 times. However, from QoS perspective, this is not relevant, such high video demands may cause the player to crash anyway. From QoE perspective this is either not relevant, since the user is already annoyed when a few stalling events happen (see Section 3).

2.3.3 Stalling Length

Next, we take a closer look at the length L of single stalling events. For each video clip, we measured the durations of each stalling event. Then, we computed several statistical measures per video clip, including mean and median of the stalling length over the stalling events of an individual clip. However, we found no correlation between the statistical measures of the stalling time and any other variable, i.e. video frame rate, stalling frequency, video bit rate, video duration, number of stallings.

Figure 5 shows the CDF of the median and the mean stalling length for the two different network capacities B. It can be seen that the curves for the mean stalling length differ with B. Nevertheless, the minimum of the average stalling length is about 2 s and for most videos the mean stalling length is below 6 s. However, there are several videos which show an even larger mean stalling length. A closer look at the individual application level stalling traces revealed that this large average stalling length was mostly caused by one large single stalling event during the playout of the individual video clip. These video clips correspond to the outliers as identified for the stalling frequency in Figure 4.

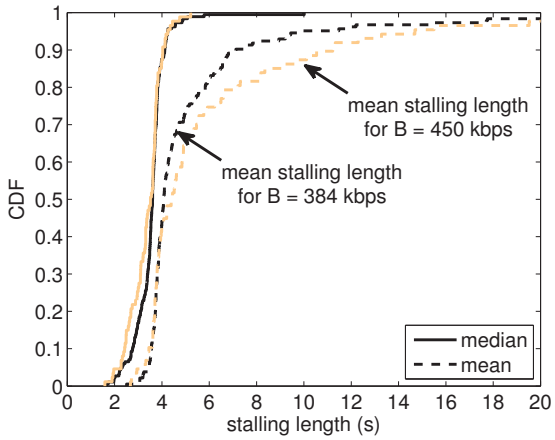

Figure 5: Median and mean of the stalling length for two different bottleneck capacities

We therefore take a closer look at the median of the stalling length to attenuate the impact of large single stalling events. In that case, the CDFs of the median of the stalling length for the two different network capacities are very close together and no impact of the bottleneck capacity on the median can be observed. In particular, the observed stalling lengths are mainly between 2 s and 5 s. Because of this observation and no correlations with other variables, we conclude that the implementation of the video playout buffer determines mainly the stalling length.

For the two bottleneck capacities, only small variances of the stalling length are observed. The coefficient of variation C can be fitted by a lognormal distribution with parameters $\mu_C = 0.786$ and $\sigma_C = 0.417$; and $\mu_C = 0.642$ and $\sigma_C = 0.402$ for $B = 384\,\text{kbps}$ and $B = 450\,\text{kbps}$, respectively.

Summarizing this section, the stalling pattern of a video can be described by stalling frequency F and stalling length L. The stalling frequency is determined by the ratio of video bit rate and bottleneck capacity. The length of a single stalling event is in the order of a few seconds and lies between 2 s and 6 s mainly.

3 Subjective User Study on YouTube Video Delivery via Transmission Control Protocol

Due to the current lack of QoE models that identify key influence factors for YouTube (e.g. demographics of users, Internet application usage habits, content types, network impairments) and explicitly address stalling effects in the context of TCP-based online video, subjective user studies need to be performed. Such studies are typically carried out by a test panel of real users in a laboratory environment. While many and possibly even diverging views on the quality of the media consumption can be taken into account – entailing accurate results and a good understanding of the QoE and its relationship with QoS – lab-based user studies can be time-consuming and costly, since the tests have to be conducted by a large number of users to obtain statistically relevant results. Because of the costs and time demands posed by laboratory tests, only a limited set of influence factors can be tested per test session. In related work [14], the correlation between network QoS in terms of delay, packet loss and throughput, application QoS in term of stalling, and QoE was evaluated for HTTP video streaming in a lab test. However, only a single video clip was used and for each test condition only ten subjects rated their experienced quality. Therefore, [14] is quite limited with respect to reliability and QoE influence factors, e.g. video content type, resolution, etc., under investigation. Costs and time demands further increase if the design and the execution of the tests as well as the analysis of the user ratings are performed in an iterative way. This means that the YouTube QoE model is developed through repeated cycles of design, implementation, and statistical analysis of the tests. This iterative approach is unavoidable when touching new QoE aspects like stalling effects.

Crowdsourcing therefore seems to be an appropriate alternative approach for deriving a QoE model. Crowdsourcing means to outsource a task (like video quality testing) to a large, anonymous crowd of users in the form of an open call. Crowdsourcing

platforms in the Internet, like Amazon Mechanical Turk or Microworkers, offer access to a large number of internationally widespread users in the Internet and distribute the work submitted by an employer among the users. The work is typically organized at a finer granularity and large jobs (like a QoE test campaign) are split into cheap (micro-)tasks that can be rapidly performed by the crowd.

With crowdsourcing, subjective user studies can be efficiently conducted at low costs with adequate user numbers for obtaining statistically significant QoE scores [5]. In addition, the desktop-PC based setting of crowdsourcing provides a highly realistic context for usage scenarios like online video consumption. However, reliability of results cannot be assumed because of the anonymity and remoteness of participants (cf. [6] and references therein): some subjects may submit incorrect results in order to maximize their income by completing as many tasks as possible; others just may not work correctly due to lack of supervision. Therefore, it is necessary to develop an appropriate methodology that addresses these issues and ensures consistent behavior of the test subjects throughout a test session and thus obtain reliable QoE results.

In order to derive the YouTube model, three steps are proposed. (1) Subjective user studies are designed which take into account the features of crowdsourcing. (2) The user studies are conducted in which several influence factors on the user perceived quality are varied. The network conditions are emulated such that the users experience a predefined stalling pattern. (3) The test results are statistically analzed in order to quantify YouTube QoE in a statistical robust way.

In the following, Section 3.1 gives a background on crowdsourcing and the platform used in this work. The subjective test methodology is presented in Section 3.2 aiming at an appropriate test design to detect unreliable user ratings. In Section 3.3, the test results are statistically analyzed. In particular, we apply different results filtering levels and assess the reliability of the data set. The YouTube QoE is then quantified in Section 3.4 for a realistic impairment scenario, where the YouTube video is streamed over a bottleneck link.

3.1 Crowdsourcing and Microworkers Platform

Crowdsourcing can be understood as a further development of the outsourcing principle by changing the granularity of work [15] and the size of the outsourced tasks, as well as the administrative overhead. A microtask can be accomplished within a few minutes to a few hours and thus does not need a long term employment. Further, it is irrelevant to the employer who actually accomplishes the task and usually the task has to be repeated several times. The repetitive tasks are combined in a *campaign*, which the employer submits to the crowdsourcing platform. The workforce in the crowdsourcing approach is not a designated worker but a large, anonymous human crowd of workers. The *crowdsourcing platform* acts as a mediator between the employer and the crowd.

In this work, we use the Microworkers[1] crowdsourcing platform, since Microworkers allows to conduct online user surveys like our YouTube QoE tests. Microworkers supports

[1] http://www.microworkers.com

workers internationally in a controlled fashion, resulting in a realistic user diversity well-suited for QoE assessment. The Microworkers platform had about 80,000 registered users end of 2010 (see [16] providing also a detailed analysis of the platform and its users).

The life cycle of a campaign in the Microworkers platform comprises the following steps. (1) First, the employer submits a campaign to the crowdsourcing platform. This includes a description of the task, the payment per task, how the workers have to proof a completed task, and how many tasks n are needed. (2) Then an employer of Microworkers reviews the campaign and approves it, if it corresponds to their guidelines. (3) Afterwards, the workers start working on the campaign and submit their finished tasks. (4) As soon as the desired n tasks are completed, the campaign is paused. The employer has to review the submitted tasks within 7 days, if they are valid. If m tasks are not valid, the campaign resumes until m new tasks are submitted. (5) If the employer rated n tasks valid, the campaign is completed. The most critical part of using crowdsourcing for subjective user tests is step (4), since it is non-trivial to decide for a subjective test, whether the task result is valid or not.

In general, every crowdsourcing task suffers from bad quality results. Therefore, different task design strategies have been proposed to improve the quality of the output. Using the example of an image labeling task, Huang et al. [17] demonstrated that the results quality of a task can be influenced by its design. They varied the payment per task, the number of requested tags per image, the number of images per task and the tasks per campaign in order to maximize the number of unique labels or the number of labels corresponding with their gold standard.

However, even if the task is designed effectively, workers might still submit incorrect work. Thus, tasks can be equipped with verification questions [18] to increase the quality, the workers input can be rechecked by others as e.g. in [19, 20], or iterative approaches can be used [21, 22]. If the workers input is not wrong but only biased, there also exist methods to eliminate these biases [23]. Based on these insights and suggestions, we developed a new, improved QoE assessment method for crowdsourcing.

3.2 Subjective Crowd Test Methodology

The test methodology developed throughout this work allows experimenters to conduct subjective user tests about the user perceived quality of Internet applications like YouTube by means of crowdsourcing and to evaluate the impact of network impairments on QoE. For the necessary quality assurance of the QoE test results themselves including the identification of unreliable user ratings, we apply different task design methods (cf. Section 3.2.1), before the subjective uers tests are conducted by the crowd (cf. Section 3.2.2). Different user study campaigns are designed (cf. Section 3.2.3) according to the influence factors under investigation.

3.2.1 Task Design Based Methods

The task design methods described in the following paragraphs can be used for different crowdsourcing tasks. Nonetheless, we describe their applicability in the context of evaluating the QoE for YouTube videostreaming.

Gold Standard Data The most common mechanism to detect unreliable workers and to estimate the quality of the results is to use questions whereof the correct results are already known. These gold standard questions are interspersed among the normal tasks the worker has to process. After results submission by the worker, the answers are compared to gold standard data. If the worker did not process the gold standard questions correctly, the non-gold standard results should be assumed to be incorrect too.

Since for subjective quality testing personal opinions are asked for, the gold standard data approach has to be applied with care since user opinions must be allowed to diverge. Still, in our tests we included videos without any stalling and additionally asked participants: "Did you notice any stops to the video you just watched". If a user then noticed stops, we disregarded his ratings for quantification of QoE. We additionally monitored the stalling events on application layer to exclude any unwanted stops, see Section 3.2.2.

Consistency Tests In this approach, the worker is asked the same question multiple times in a sightly different manner. For example, at the beginning of the survey the worker is asked how often she visits the YouTube web page, at the end of the survey she is asked how often she watches videos on YouTube. The answers can slightly differ but should be lie within the same order of magnitude. Another example is to ask the user about his origin country in the beginning and about his origin continent at the end. The ratings of the participant are disregarded, if not all answers of the test questions are consistent.

Content Questions After watching a video, the users were asked to answer simple questions about the video clip. For example, "Which sport was shown in the clip? A) Tennis. B) Soccer. C) Skiing." or "The scene was from the TV series... A) Star Trek Enterprise. B) Sex and the City. C) The Simpsons." Only correct answers allow the user's ratings to be considered in the QoE analysis.

Mixed Answers This method is an extension to consistency tests to detect workers using fixed click schemes in surveys. Usually, the rating scales on surveys are always structured in the same way, e.g. from good to bad. Consequently, workers using fixed click scheme might bypass automated consistency tests, as always selecting the first or the middle answer results in a consistent survey. An easy way to avoid this is to vary the structure of the rating scales. For example the options of the first quality question "Did you notice any stops while the video was playing?" has the order "No", "Yes", whereas in the following question "Did you experience these stops as annoying?" the order is "Extremely","Fairly",..., "Not at all". Now, following a fixed clicking scheme results causes inconsistencies and identifies unreliable participants.

Application Usage Monitoring Monitoring users during the tasks completion can also be used to detect cheating workers. The most common approach here is measuring the time the worker spends on the task. If the worker completes a task very quickly, this might indicate that she did the work sloppy.

In this work, we monitored browser events in order to measure the focus time, which is the time interval during which the browser focus is on the website belonging to the user test. In order to increase the number of valid results from crowdsourcing, we displayed a warning message if the worker did not watch more than 70 % of the video. The users could decide to watch the video again or to continue the test. When workers became aware of this control mechanism, the percentage of completely watched videos doubled and almost three times more workers could be considered reliable than without the system warning.

For the subjective crowd tests, we recommend to combine all above mentioned task designs, i.e. gold standard data, consistency checks, content questions, mixed questions and application monitoring.

3.2.2 Implementation and Execution of Experiments

The aim of the experiments is to quantify the impact of network impairments on QoE. For YouTube video streaming, network impairments result into related stalling patterns. As the video experience should be as similar as possible to a visit of the real YouTube website, the application should run on the users' default web browser. In order to provide dynamic web content the application was based on JavaServer Pages (JSP). The JSPs are compiled into servlets that are able to receive and respond to HTTP requests. These servlets ran on an Apache Tomcat server set up on Debian GNU/Linux. This server included a MySQL relational database for logging test settings, user events and answers.

To this end, an instance of the YouTube Chromeless Player was embedded into dynamically generated web pages. With JavaScript commands the video stream can be paused, a feature we used to simulate stalling. In addition, the JavaScript API allows to monitor the player and the buffer status, i.e. to monitor stalling on application layer. In order to avoid additional stalling caused by the test users' Internet connection, the videos had to be downloaded completely to the browser cache before playing. This enables us to specify fixed unique stalling patterns which are evaluated by several users.

During the initial download of the videos, a personal data questionnaire was completed by the participant which also includes consistency questions from above. The user then sequentially viewed three different YouTube video clips with a predefined stalling pattern. After the streaming of the video, the user was asked to give his current personal satisfaction rating during the video streaming. In addition, we included gold standard, consistency, content and mixed questions to identify reliable subjective ratings. The workers were not aware of these checks and were not informed about the results of their reliability evaluation. Users had to rate the impact of stalling during video playback on a 5-point absolute category rating (ACR) scale [24] with the following values: (1) bad; (2) poor; (3) fair; (4) good; (5) excellent.

3.2.3 Design of Campaigns with Respect to Influence Factors

For deriving the impact of various influence factors, we conducted individual crowdsourcing campaigns in which only a single parameter is varied, while the others are kept constant. This strict separation helps for a proper QoE analysis and deriving adequate QoE models. In this work, we focus on the quantification of network impairments on YouTube QoE. Since YouTube videos are delivered via TCP, any network impairments appear as stalling to the end user.

For obtaining realistic stalling patterns we first studied the relationship between network QoS and stalling events. To this end, several YouTube videos were requested with a downlink bandwidth limitation of the used browser. On network layer, packet traces were captured, while on application layer, the YouTube player status (i.e. playing or stalling) was monitored by using the YouTube Javascript API. In case of a bottleneck, i.e if the available bandwidth is lower than the video bandwidth, the video play back stalls several times. For example, we requested a 30 s Avatar trailer with an average video bitrate of 817.6 kbps or 102.2 kBps. We varied the bottleneck bandwidth b between 20 kBps and 102.2 kBps. As a result, we found that the stalling events occur periodically. For the example trailer, the number N of stallings can be approximated by $N(b) = \max\{-0.467 \cdot b + 27.616, 0\}$, while the total stalling time T follows as $T(b) = \max\{1\,237e^{2.323/x} - 1\,286, 0\}$. The average length L of a single stalling event follows as $L(b) = {}^{T(b)}/_{N(b)}$. We found that for our videos, a bandwidth of about 60 kBps was sufficient to play out the video without any interruptions, since an initial buffering process prevents stalling in this case. Details can be found in the technical report [25].

As a result of this analysis, we parametrized our crowdsourcing campaigns $C1 - C7$ as outlined in Table 3, varying either length or number of stalling events while keeping the other parameter constant.

id	number N of stallings	length L of stalling event
$C1$	$0, 1, 2, 3, 4, 5, 6$	4 s
$C2$	1	$2, 4, 8, 16, 32, 64$ s
$C3$	$0, 1, 2, 3, 4, 5, 6$	1 s
$C4$	$0, 1, 2, 3, 4, 5, 6$	2 s
$C5$	2	$1, 2, 4, 8, 16, 32$ s
$C6$	3	$1, 2, 4, 8, 16$ s
$C7$	$0, 1, 2, 3, 4, 5, 6$	3 s

Table 3: Parametrization of the seven crowdsourcing campaigns

3.3 Statistical Analysis of Test Results

Throughout our measurement campaign, 1 349 users from 61 countries participated in the YouTube stalling test and rated the quality of 4 047 video transmissions suffering from stalling. Statistical analysis of the demographics of the users can be found in [25]. We first identify unreliable users and filter the data from the user studies accordingly.

Then, we show that the (inter-rater and intra-rater) reliability of the filtered data is improved significantly.

3.3.1 Unreliable Users and Filtering of Data

The task design based methods as defined in Section 3.2.1 allow a three level filtering of the users. Based on the answers of the users to the consistency, content and mixed questions as well as on the application monitoring, we applied a three level filtering in order to detect the reliable workers. For all steps of the filtering process the application assisted by indicating possible cheaters. However, all these suggested cheaters data were revised manually.

The first level identifies crowdsourcing users that gave wrong answers to content questions, that provided different answers to the same rephrased consistency questions, or that often selected the same option during the test. Thus, the first level applies consistency tests, content questions and mixed answers. The second level checks additionally whether participants who watched a video with stops noticed the stalling and vice versa, i.e., gold standard data is included in the test. The third level extends the previous filter level by additionally monitoring the application usage. All users are removed that did not watch all three videos completely.

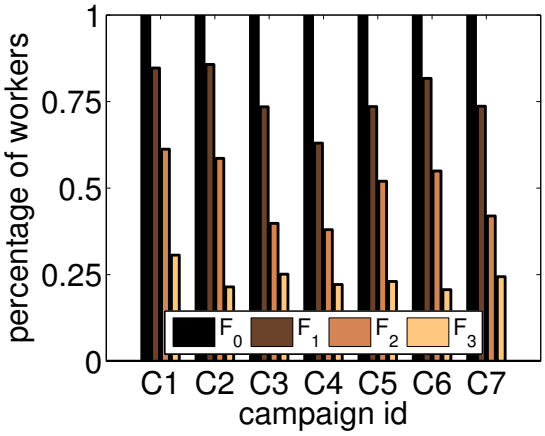

Figure 6: Percentage of remaining participants per filter level

Figure 6 shows the percentage of users passing the three filter levels for the different crowdsourcing campaigns $C1, \ldots, C7$ we performed. In each of the user study campaigns we only varied a single test condition (either the number of stallings or the duration of a single stalling event), while the remaining test conditions like video contents were kept equal. Level 0 refers to the unfiltered data from all users.

Interestingly, each filter technique reduces the number of valid crowdsourcing workers by approx. 25 % on average over all campaigns. This indicates that the consistency tests are quite useful for identifying spammers clicking random answers as well as video content questions and monitoring task specific parameters (like the focus time) for identifying

sloppy workers who do not watch the video carefully enough. Monitoring task specific parameters like the focus time on the video, also helps to identify unreliable users. However, in our case, the monitoring is implemented via Javascript, i.e. monitoring of window.onBlur and window.onFocus events, which do not seem to work correctly across all browsers[2]. Therefore, our three level filtering may be even too pessimistic, but leads to valid test data suitable for quantifying YouTube QoE. In contrast to the consistency checks and content questions, monitoring task specific parameters are much more complicate to develop and to implement, as they differ for each crowdsourcing task. Due to our restrictive filtering, only about one fourth of the subjective ratings were finally considered for the analysis.

3.3.2 Reliability of Filtered Data

We consider two different types of reliability of the user studies: intra-rater and inter-rater reliability. Firstly, *intra-rater reliability* determines to which extent the ratings of an individual user are consistent. In a measurement campaign C, an individual user u sequentially views three different YouTube video clips with a predefined stalling pattern x_i for $i \in \{1, 2, 3\}$ and rates the QoE accordingly with y_i. In each campaign, we only vary a single test condition (either the number of stalling pattern or the length of a single stalling event) and keep the others constant. Hence, we assume that worse stalling conditions $x_j > x_k$ will be reflected accordingly by the the user ratings with $y_j \leq y_k$. Therefore, we can apply the Spearman rank-order correlation coefficient $\rho_{C;u}(x_u, y_u)$ for ordinal data between the user ratings y_u and the varied stalling parameter x_u. Spearman rank correlation considers only that the items on the rating scale represent higher vs. lower values, but not necessarily of equal intervals. We define the intra-rater reliability ρ_C of a campaign C by averaging over all users $\mathcal{U}$, i.e. $\rho_c = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \rho_{C,u}$.

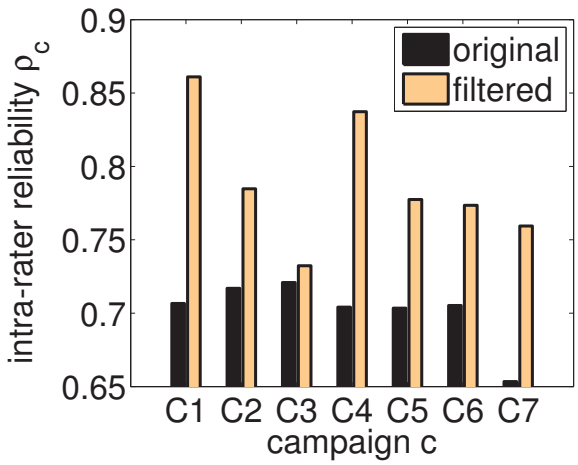

Figure 7: Increase of intra-rater reliability of filtered data compared to original data

[2]See `http://www.quirksmode.org/dom/events/`

Secondly, *inter-rater reliability* denotes the degree of agreement among raters. For a campaign c, we define it as Spearman rank-order correlation coefficient κ_c between all user ratings y_C and the varied stalling parameter x_C for all user ratings in a campaign. It has to be noted that the applied filter levels are independent of the actual stalling conditions, hence, the above defined reliability metrics are valid.

Figure 7 and Figure 8 shows the intra-rater reliability ρ_C and the inter-rater reliability κ_C of the different campaigns for the original data and the filtered data applying level 3, respectively. It can be seen that the intra- and inter-rater reliability is increased in all campaigns, thus, the filtering succeeds in identifying unreliable users. On average, ρ_C and κ_C is increased about 0.0879 and 0.2215, respectively. The three level filtering of the users from campaign $C3$ only leads to a slight increase of the intra-rater reliability. This is due to the fact that $C3$ investigates the influence of very short stallings of length 1 s and it seems to be more difficult for individual users to rate the influence on the 5-point ACR scale appropriately. Nevertheless, the inter-rater reliability of campaing $C3$ is significantly improved. The inter-reliability of campaign $C2$ is lower than in the other user study campaigns, since we consider very long stalling events up to 64 s within $C2$.

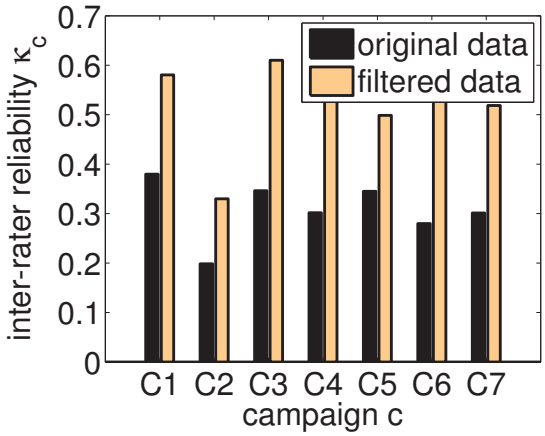

Figure 8: Increase of inter-rater reliability of filtered data compared to orignial data

3.4 Quantification of YouTube QoE

The quantification of YouTube QoE aims at inferring the subjective user rating from the stalling parameters. This includes an analysis of the user diversity conducted by means of the SOS hypothesis, before the key influence factors on YouTube QoE are investigated. Finally, a mapping between the user ratings and the key influence factors are presented. Together with the quantification of user diversity, the mapping function provides a complete picture of YouTube QoE.

3.4.1 User Diversity and the SOS Hypothesis

The reliability of the data indicates to which extent the users give consistent QoE ratings. However, a certain heterogenity of the test subjects' opinions on the quality experienced remains, caused by several psychological influence factors such as individual expectations regarding quality levels, type of user and sensitivity to impairments, uncertainty how to rate a certain test condition, etc. Therefore, we investigate this diversity among users and show that the filtered data leads to valid results.

To this end, we analyze quality ratings where users experience the same individual test conditions, i.e. the same number of stalling events and the same length of single stalling events. The SOS hypothesis as introduced in [26] postulates a square relationship between the average user ratings $\mathrm{MOS}(x)$ and the standard deviation $\mathrm{SOS}(x)$ of the user ratings for the same test condition x: $\mathrm{SOS}(x)^2 = a\left(-\mathrm{MOS}(x)^2 + 6 \cdot \mathrm{MOS}(x) - 5\right)$. Then, the SOS parameter a is characteristic for certain applications and stimuli like waiting times. Web surfing is closely related to YouTube videostreaming due to the TCP-based delivery of data and the resulting waiting times due to network impairments. For web surfing, the SOS parameter is about 0.3 according to [26].

For the unfiltered YouTube user ratings, we obtain a SOS parameter of 0.4592 which is very large and shows an even larger user diversity than for complex cloud gaming [27]. Thus, the unfiltered data do not seem to be valid from this perspective. Considering the filtered data, we obtain an SOS parameter of 0.3367 which lies in the range of web surfing. This clearly indicates the validity of the filtered data. Consequently, we consider only filtered data in the following because of their reliability and validity.

3.4.2 Key Influence Factors on YouTube QoE

In the crowdsourcing campaigns, we focused on quantifying the impact of stalling on YouTube QoE and varied 1) the number of stalling events as well as 2) the length of a single stalling event, resulting in 3) different total stalling times. We also considered the influence of 4) the different crowdsourcing campaigns, 5) the test video id in order to take into account the type of video as well as the resolution, used codec settings, etc. Further, we asked the users to additionally rate 6) whether they liked the content (using a 5-point ACR scale). We collected additional data concerning the *demographics* of the user by integrating demographical questions in the survey. In particular, we asked the users about their 6) age, 7) gender, 8) family situation, 9) education, 10) profession, 11) home country, 12) and home continent.

To get insights into the users expectations and habits in the context of YouTube, we additionally estimated 13) the user's access speed by measuring the time for downloading the video contents. Further, 14) the used browser was monitored by reading the user-agent field in the HTTP request header. Finally, we asked the users how their 15) YouTube usage and 16) Internet usage, i.e. how often the use YouTube or the Internet (several times per day, once a day, several times per week, once a week, several times per month, less often, never).

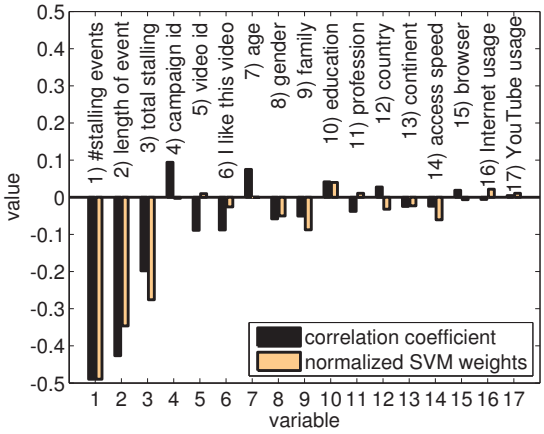

Figure 9: Identification of key influence factors on YouTube QoE

Finally, the key influence factors on YouTube QoE are identified by means of (a) correlation coefficients and (b) support vector machine (SVM) weights. We compute the Spearman rank-order correlation coefficient between the subjective user rating and the above mentioned variables. In addition, we utilize SVMs as machine learning approach to make a model for classification. Every variable gets a weight from the model indicating the importance of the variable. However, SVMs are acting on two-class problems only. For this, we take the categories 1 to 3 of the ACR scale to class "bad quality" and the categories 4 to 5 to class "good quality". We choose the implementation of SMO (Sequential Minimal Optimization [28]) in WEKA [29] for analysis.

Figure 9 shows the results from the key influence analysis. On the x-axis, the different influence factors ν_i are considered, while the y-axis depicts the correlation coefficient α_i as well as the SVM weights β_i which are normalized to the largest correlation coefficient for the sake of readability. We can clearly observe from both measures α_i and β_i, that the stalling parameters dominate and are the key influence factors. Surprisingly, the user ratings are statistically independent from the video parameters (like resolution, video motion, type of content like news or music clip, etc.), the usage pattern of the user, as well as its access speed to reflect the user's expectations. As future work, we will further investigate such influence factors by considering more extreme scenarios (e.g. very small resolution vs. HD resolution).

3.4.3 Mapping between MOS and Stalling

The analysis in the previous subsection has shown that YouTube QoE is mainly determined by stalling and both stalling parameters, i.e. frequency and length. For quantifying YouTube QoE, concrete mapping functions depending on these two stalling parameters have to be derived.

First, we investigate the invididual user ratings in detail, cf. Figure 10. The x-axis denotes the number of stallings whereas the y-axis denotes the user ratio for the different

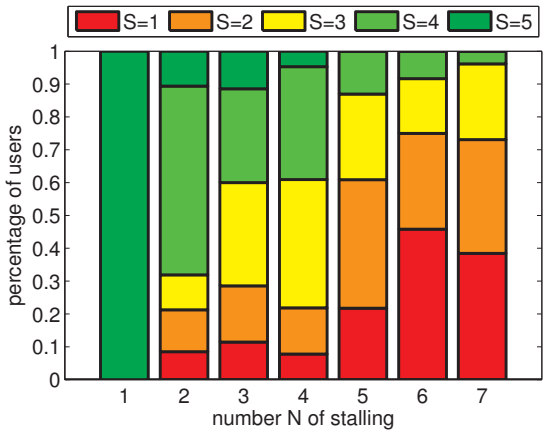

Figure 10: User ratings for length of 3 s of a single stalling event

ratings ($S = i$). It can be seen that all users rate the video play out with maximum rating $S = 5$ if no stalling occurs. Further, for one stalling event of length 3 s the full rating scale is already exploited, whereas the majority of users still rank the video playback with at least a $S = 4$, i.e. good quality. In case of two and more stalling events more than 30 % of the users rate the video experience with the lowest rating score $S = 1$. For three or more stalling events the opinion score distribution does not change significantly. This is due to the fact that it does not matter any more if the user has to wait four or more times during the video playback; the perceived quality is too low, the user is dissatisfied. We can conclude that users might excuse one stalling, but more stalling events, especially more than two, significantly reduce the user perceived video quality.

Figure 11 depicts the MOS values for one and three seconds stalling length for varying number of stalling events. In addition, the MOS values are fitted according the IQX hypothesis as discussed in [30]. The IQX hypothesis formulates a fundamentail relationship between QoE and an impairment factor corresponding to the QoS. According to the IQX hypothesis, the change of QoE depends on the current level of QoE – the expectation level– given the same amount of change of the QoS value. Mathematically, this relationship can be expressed by a differential equation

$$\frac{\partial QoE}{\partial QoS} = -\beta(QoE - \gamma) \tag{2}$$

which can be easily solved as an exponential functional relationship between QoE and QoS.

In the context of YouTube QoE for TCP based video streaming, the number of stallings is considered as impairment. Hence, QoE in terms of MOS is described by an exponential function. The mapping functions between the number N of stalling events of length L are given in Table 4 which also shows the coefficients of determination R_L^2 for the different fitting functions being close to perfect match, i.e. $R_L^2 = 1$. The results in Figure 11 show that users tend to be highly dissatisfied with two ore more stalling events per clip.

However, for the case of a stalling length of 1 s, the user ratings are substantially better for same number of stallings. Nonetheless, users are likely to be dissatisfied in case of four or more stalling events, independent of the stalling duration.

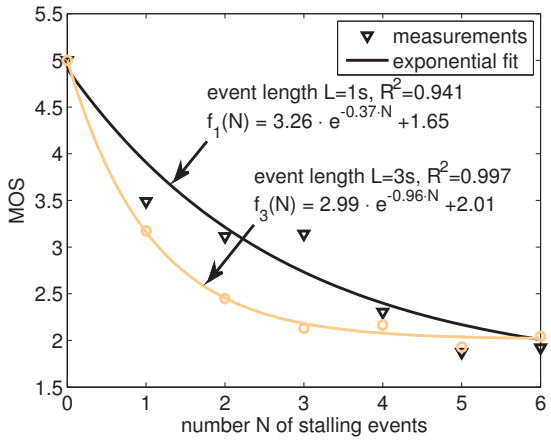

Figure 11: Mapping functions of stalling parameters to mean opinion scores

We additionally investigated whether it is possible to summarize the stalling pattern into a total stalling time $T = N \cdot L$. For this, we used the fitting functions $f_L(N)$ for the different stalling lenghts and compared the functions $h_L(T) = h_L(L \cdot N) = f_L(N)$ transformed to total stalling times. However, we found that the resulting curves significantly differ, hence, it is not sufficient to summarize the stalling pattern into total stalling times. This is inline to our findings in Section 3.4.2 where both stalling parameters revealed equal importance and thus have to be taken into account individually for any mapping functions.

length L	*mapping function* $f_L(N)$	R_L^2
1 s	$f_1(N) = 3.26 \cdot e^{-0.37 \cdot N} + 1.65$	0.941
2 s	$f_2(N) = 2.99 \cdot e^{-0.69 \cdot N} + 1.95$	0.923
3 s	$f_3(N) = 2.99 \cdot e^{-0.96 \cdot N} + 2.01$	0.997
4 s	$f_4(N) = 3.35 \cdot e^{-0.89 \cdot N} + 1.62$	0.978

Table 4: Mapping functions between MOS and number N of stalling events of length L as well as coefficient of determination for TCP transmission

4 Comparison of YouTube Quality of Experience for UDP and TCP Transport Protocols

For quantifying the influence of the transport protocol on the QoE, we consider now the bottleneck scenario with a given bottleneck capacity B. In case of TCP based video streaming, the bottleneck may lead to stalling as QoE impairment. According to our findings in Section 2 a given bottleneck link capacity results in a certain stalling pattern, i.e. a certain stalling frequency F and a certain stalling length L. With the YouTube QoE model in Section 3, the stalling pattern can then be mapped to a MOS. In case of UDP based video streaming, the bottleneck link capacity may lead to packet loss as QoE impairment. Then, the QoE model from Section 4.1 can be applied to quantify the QoE in terms of MOS for a given packet loss ratio R. Hence, in both cases, TCP or UDP based video streaming, the bottleneck link capacity is mapped to MOS. In the following, we show how this mapping is applied in case of UDP (Section 4.2) and TCP (Section 4.3). In order to have a fair comparison between UDP and TCP based transmission of video contents, we neglect any initial delays. Finally, Section 4.4 compares both protocols from the end user perspective, when the video stream is delivered over a bottleneck.

4.1 Quality Assessment of UDP-based YouTube Videostreaming

For assessing the user perceived quality of YouTube video streaming using the UDP transport protocol, we rely on a publicly available database, that is the "EPFL-PoliMI video quality assessment database" at `http://vqa.como.polimi.it/`. Its video streams are encoded with H.264, the same codec used by YouTube. Twelve different video sequences were investigated from which one half has a spatial CIF resolution (352×240 pixel) and the other half 4CIF resolution (704×480 pixel). For each of the twelve original H.264 bit-streams, a number of corrupted bit-streams were generated, by dropping packets according to a given error pattern. The error patterns were generated at six different packet loss ratios R, that are 0.1 %, 0.4 %, 1 %, 3 %, 5 %, 10 %. Furthermore, two different types of error patterns are considered, that are random errors and bursty errors. Thus, in total, 72 CIF and 72 4CIF video sequences with packet losses as well as the original 6 CIF and 6 4CIF sequences without packet losses were considered in the subjective tests.

The CIF and 4CIF video sequences were presented in two separate test sessions to the test users. At the end of each video sequence, the subjects were asked to rate the quality using a five-point ITU continuous adjectival scale. Using a slider, the test users continuously rate the instantaneously perceived quality using an adjectival scale from "bad" to "excellent", which corresponds to an equivalent numerical scale from 0 to 5. Thus, in contrast to the subjective user study in the previous section 3, "bad" quality rating y is any continuous value between 0 and 1, i.e. $0 \leq y \leq 1$, while "excellent" quality rating means $4 < y \leq 5$. In total, fourty naive subjects took part in the subjective tests. More details on the subjective test can be found in [31, 4].

Figure 12 shows the MOS depending on the simulated packet loss ratio R for the two different resolutions CIF and 4CIF. For each packet loss ratio R and each video

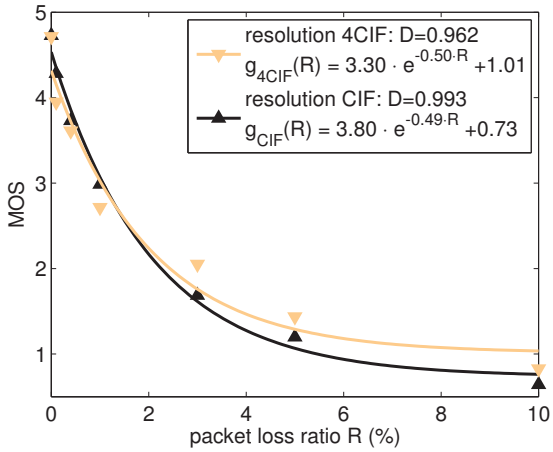

Figure 12: MOS values and mapping function between packet loss ratio R for UDP based streaming

resolution, the subjective ratings from all test users (across the different video contents and the type of error pattern) were averaged to obtain the corresponding MOS value. It can be seen that the MOS strongly decays with increasing network impairment in terms of packet loss.

To this end, we consider the packet loss ratio as impairment factor on the QoE. Hence, we can apply again the IQX hypothesis in order to derive a mapping function between the QoS impairment, i.e. the packet loss ratio, and the QoE in terms of MOS. As a result, we obtain an exponential mapping function between QoE and QoS which is depicted as solid line in Figure 12. Furthermore, the mapping function itself is shown in the plot. Again, we see a very good match of the mapping function and the measured MOS values which is quantified by the coefficient of determination being close to a perfect match.

As a result, we see that in the case of UDP-based video streaming, packet loss is a key influence factor on QoE. In contrast, the resolution of the video contents (CIF vs. 4CIF) has only a minor impact on the MOS.

4.2 UDP based Streaming with Packet Loss

During the video of length D, about $\frac{D \cdot B}{S}$ packets of size S are downloaded with a download bandwidth B. Since the video (encoded with bitrate V) consists of $\frac{D \cdot V}{S}$ packets, the packet loss ratio follows as

$$R = 1 - \frac{B}{V} \,. \tag{3}$$

Accordingly, the mapping Υ_v between the normalized throughput $\rho = \frac{B}{V}$ and the MOS value is derived as

$$\Upsilon_v(\rho) = f_v\,(1 - \rho) \tag{4}$$

using the mapping function $f_v(R)$ between the packet loss ratio R and the MOS value as defined in Section 4.1 for a given video resolution v.

4.3 TCP based Video Streaming with Stalling

The download time T_d of a video of duration D which is encoded with average video bitrate V depends on the capacity B of the bottleneck,

$$T_d = \frac{V \cdot D}{B} \,. \tag{5}$$

Thus, the total stalling time T_s follows as difference $T_d - D$ between the download time and the video duration,

$$T_s = \left(\frac{V}{B} - 1 \right) D \,. \tag{6}$$

Then, the number N of stalling events of length L is

$$N = \left(\frac{V}{B} - 1 \right) \frac{D}{L} = \left(\frac{1}{\rho} - 1 \right) \frac{D}{L} \,. \tag{7}$$

Together with the normalized throughput ρ which is defined as the ratio between the bandwidth limitation B and the video bitrate V, i.e. $\rho = \frac{B}{V}$, we arrive at the following mapping function Υ_L between the normalized throughput and the MOS value,

$$\Upsilon_L(\rho) = f_L \left(\left(\frac{1}{\rho} - 1 \right) \frac{D}{L} \right) \,, \tag{8}$$

where $f_L(N)$ is defined as in Section 3 in Figure 11 or Table 4.

In addition to this simple model for obtaining the stalling pattern to a given bottleneck capacity B, we can use the fitting function in Eq.(1) which returns the stalling frequency $F = {}^{N}/_{D}$ for given ${}^{V}/_{B} = {}^{1}/_{\rho}$.

4.4 Comparison of User Perceived Quality for TCP and UDP based YouTube Video Delivery

In this section, we combine the results from the previous subsections in order to compare the QoE for YouTube video streaming over a bottleneck with capacity B. For TCP based transmission, this results in stalling which degrades the QoE; for UDP based transmission, the bottleneck results into packet loss and corresponding visual impairments of the video.

Thus, for the current two Internet protocols, TCP and UDP, the same QoS impairment in terms of the bottleneck bandwidth will lead to completely different QoE impairments. Thus, it is possible to evaluate which kind of stalling pattern (in terms of number of stallings and length of a single stalling event) corresponds to which packet loss ratio, such that the user experiences the same QoE. Figure 13 shows the number N of stallings on the x-axis and the corresponding packet loss ratio R on the y-axis which result in the same MOS value, which is indicated by the color of the point. Two different curves are depicted according to a stalling length of $L = 1\,\mathrm{s}$ and $L = 4\,\mathrm{s}$. For the mapping between packet loss and MOS we used the CIF resolution. For example, $N = 2$ stallings of length $L = 4\,\mathrm{s}$ correspond to a packet loss ratio $R = 2\,\%$ and lead to a MOS value about 2, i.e.

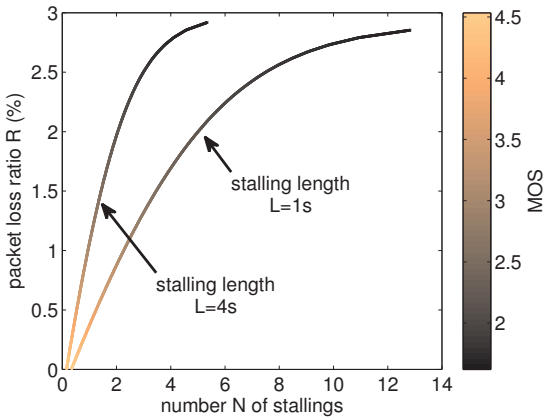

Figure 13: MOS color plot wrt. stalling frequency (TCP) and packet loss ratio (UDP)

bad quality. It can be seen, that the transformation between both impairment factors is quite complex and non-linear.

Finally, we compare both protocols, TCP and UDP, for a given bottleneck bandwidth B in terms of MOS. In particular, we use the normalized throughput ρ as ratio of the bottleneck bandwidth B and the video bitrate V. Then, we can directly use the mapping functions in Eq.(8) and in Eq.(4) based on the subjective user studies presented in Section 3 and in Section 4.1 for TCP and UDP, respectively.

Figure 14 shows the numerical results depending on the normalized throughput ρ. In case of TCP, we use the mapping functions based on the four different stalling length from $L = 1\,\mathrm{s}$ to $L = 4\,\mathrm{s}$. In addition, the measurement results from Section 2.3 are used. For the different videos streamed over a bottleneck, we measured the video bitrate, the duration of the video, the observed number of stallings, and the median of the stalling length. These values are used as input in Eq.(8) to obtain a MOS value. The first observation is that the measured stalling values mapped to MOS are in the range of the curves $\Upsilon_L(\rho)$, although the assumptions in Section 4.3 are quite rough and neglect aspects like variable bitrate or initial video buffer time.

In case of UDP, the MOS values are plotted for the CIF and the 4CIF resolution with respect to ρ in Figure 14. The second observation is that UDP always performs worse than TCP from the end user perspective. Hence, for the same bottleneck capacity, the end user will likely more tolerate the resulting stalling in case of TCP than the resulting video quality degradation in case of UDP.

The results indicate that TCP based video streaming actually used by YouTube outperforms UDP based video streaming in terms of user perceived quality for network bottleneck scenarios. However, it has to be noted that also techniques for overcoming the video quality degradation due to packet losses in case of UDP do exist. By allowing buffering as well as additional retransmission mechanisms on the application layer, UDP based streaming approach might be enhanced significantly and even keep up with TCP. Furthermore, we have restricted the results of this paper to the bottleneck sce-

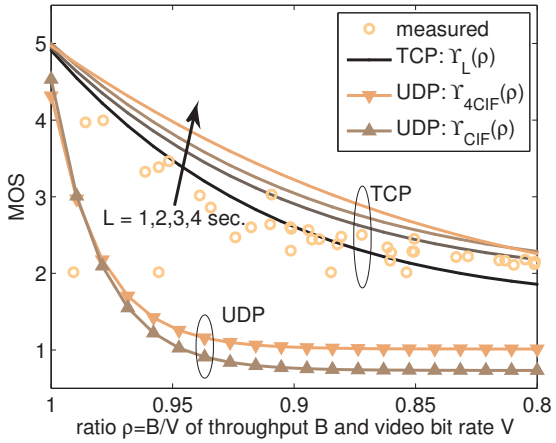

Figure 14: Comparison of UDP and TCP streaming for different bottleneck capacities (CIF resolution)

nario. Therefore, it would be interesting to investigate if the results can be transfered to lossy links scenarios or if UDP might be the appropriate choice for such scenarios, as the TCP throughput is approximately proportional to $1/\sqrt{R}$, cf. [32]. In addition, an investigation of other transport protocols like DCCP and SCTP would reveal their ability for video streaming and identify the optimal transport protocol for a YouTube like streaming service.

5 Conclusions and Outlook

Quality of Experience as a subjective measure of the end-customer's quality perception has become a key concept for analyzing Internet applications like YouTube video streaming from the end user's perspective. Therefore, in this technical report we have taken a closer look at the impact of the current Internet transport protocols on QoE for YouTube video streaming. In particular, we have investigated the quality degradations which occur in case of network bandwidth bottlenecks in case of TCP and UDP based video streaming.

For UDP based video streaming, a network bottleneck may result into packet loss and therefore visual impairments of the video contents. In contrast, TCP based video streaming, as currently implemented by YouTube, will not suffer from video quality degradation, i.e. the video content itself is not disturbed, however the bottleneck may lead to stalling of the video stream. The question arises which of both protocols is more appropriate in case of a bottleneck from the end user's perspective.

Therefore, we conducted a large-scale measurement study of YouTube video streaming over a bottleneck, in order to derive and model the resulting stalling pattern. This stalling pattern is non-trivial, due to a number of interactions and correlations on several layers of the ISO/OSI stack. YouTube implements flow control on application layer;

TCP implements flow control on transport layer; the video player implementation tries to overcome stalling by means of a video buffer; and the videos are encoded with variable bit rates. However, we found that the stalling patterns can be modeled in the following way: the stalling frequency as ratio of the number of stallings and the video duration simply depends on the normalized video demand, which is the ratio of the video bit rate and the bottleneck link capacity. However, their relation follows a non-linear exponential function. The median of the length of a single stalling event was found to be between two seconds and four seconds. With these two parameters, the observed stalling pattern can be modeled for a given bottleneck bandwidth.

As second contribution, we presented the results of two subjective user studies from literature and transformed them accordingly in order to predict user perceived quality for a given bottleneck bandwidth. The first subjective measurement campaign considers QoE when stalling occurs in case of TCP video streaming. The second subjective measurement study allows to quantify QoE when packets get lost in case of UDP video streaming. Finally, this allows to compare the influence of UDP and TCP in the bottleneck scenario. Our results show that TCP outperforms UDP for any given bottleneck bandwidth. Furthermore, we have seen that some basic considerations regarding the observed stalling pattern also enable accurate results in terms of predicted QoE.

Due to the lack of YouTube QoE models, we have quantified QoE of YouTube on behalf of the results of seven crowdsourcing campaigns. We have shown that for this application, QoE is primarily influenced by the frequency and duration of stalling events. In contrast, we did not detect any significant impact of other factors like age, level of internet usage or content type. Our results indicate that users tolerate one stalling event per clip as long as stalling event duration remains below 3 s. These findings together with our analytical mapping functions that quantify the QoE impact of stalling can be used as guidelines for service design and network dimensioning.

Furthermore, we demonstrated how crowdsourcing can be used for fast and scalable QoE assessment for online video services, since testing is parallelized and campaign turnaround times lie in the range of a few days. We also showed that results quality are an inherent problem of the method, but can be dramatically improved by filtering based on additional test design measures, i.e. by including consistency, content, and gold standard questions as well as application monitoring. Albeit such filtering can result in a 75 % reduction of user data eligible for analysis, crowdsourcing still remains a cost-effective testing method since users are typically remunerated with less than 1 $. For these reasons we believe that crowdsourcing has high potential not only for testing online video usage scenarios, but also for QoE assessment of typical Internet applications like web surfing, file downloads and cloud gaming.

References

[1] Cisco Systems Inc., "Cisco Visual Networking Index: Forecast and Methodology, 2010-2015," June 2011.

[2] M. Shiels, "YouTube at five- 2 bn views a day ." `http://news.bbc.co.uk/2/hi/technology/~8676380.stm`, 2011.

[3] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On dominant characteristics of residential broadband internet traffic," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, (New York, NY, USA), pp. 90–102, ACM, 2009.

[4] F. De Simone, M. Tagliasacchi, M. Naccari, S. Tubaro, and T. Ebrahimi, "H.264/AVC video database for the evaluation of quality metrics," in *Proceedings of the 35th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010)*, 2010.

[5] K. Chen, C. Chang, C. Wu, Y. Chang, C. Lei, and C. Sinica, "Quadrant of Euphoria: A Crowdsourcing Platform for QoE Assessment," *IEEE Network*, vol. 24, Mar. 2010.

[6] M. Hirth, T. Hoßfeld, and P. Tran-Gia, "Cost-Optimal Validation Mechanisms and Cheat-Detection for Crowdsourcing Platforms," in *Workshop on Future Internet and Next Generation Networks*, (Seoul, Korea), Jun. 2011.

[7] S. Alcock and R. Nelson, "Application flow control in youtube video streams," *SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 24–30, April 2011.

[8] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: a platform for high-performance internet applications," *SIGOPS Oper. Syst. Rev.*, vol. 44, pp. 2–19, August 2010.

[9] T. Mori, R. Kawahara, H. Hasegawa, and S. Shimogawa, "Characterizing traffic flows originating from large-scale video sharing services.," in *Traffic Monitoring and Analysis* (F. Ricciato, M. Mellia, and E. W. Biersack, eds.), vol. 6003 of *Lecture Notes in Computer Science*, Springer, 2010.

[10] V. Adhikari, S. Jain, and Z. Zhang, "Where do you tube? uncovering youtube server selection strategy," in *IEEE ICCCN 2011*, July 2011.

[11] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*, pp. 229 –238, june 2008.

[12] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, (New York, NY, USA), pp. 15–28, ACM, 2007.

[13] T. Hoßfeld and K. Leibnitz, "A Qualitative Measurement Survey of Popular Internet-based IPTV Systems," in *Second International Conference on Communications and Electronics (HUT-ICCE 2008)*, (Hoi An, Vietnam), June 2008.

[14] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of http video streaming," in *IEEE/IFIP IM (Pre-conf Session)*, (Dubland, Ireland), May 2011.

[15] T. Hoßfeld, M. Hirth, and P. Tran-Gia, "Modeling of Crowdsourcing Platforms and Granularity of Work Organization in Future Internet," in *International Teletraffic Congress (ITC)*, (San Francisco, USA), Sept. 2011.

[16] M. Hirth, T. Hoßfeld, and P. Tran-Gia, "Anatomy of a crowdsourcing platform - using the example of microworkers.com," in *Workshop on Future Internet and Next Generation Networks*, (Seoul, Korea), June 2011.

[17] E. Huang, H. Zhang, D. Parkes, K. Gajos, and Y. Chen, "Toward Automatic Task Design: A Progress Report," in *ACM SIGKDD Workshop on Human Computation*, (Washington, USA), July 2010.

[18] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing User Studies with Mechanical Turk," in *ACM SIGCHI Conference on Human Factors in Computing Systems*, (Florence, Italy), Apr. 2008.

[19] L. Von Ahn and L. Dabbish, "Labeling Images with a Computer Game," in *ACM SIGCHI Conference on Human Factors in Computing Systems*, (Vienna, Austria), Apr. 2004.

[20] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "re-CAPTCHA: Human-Based Character Recognition via Web Security Measures," *Science*, vol. 321, Sept. 2008.

[21] G. Little, L. Chilton, M. Goldman, and R. Miller, "Turkit: Tools for Iterative Tasks on Mechanical Turk," in *ACM SIGKDD Workshop on Human Computation*, (Paris, France), June 2009.

[22] P. Dai, Mausam, and D. S. Weld, "Decision-Theoretic Control of Crowd-Sourced Workflows," in *24th. AAAI Conference on Artificial Intelligence*, (Atlanta, USA), July 2010.

[23] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality Management on Amazon Mechanical Turk," in *ACM SIGKDD Workshop on Human Computation*, (Washington, DC, USA), July 2010.

[24] International Telecommunication Union, "Subjective video quality assessment methods for multimedia applications," *ITU-T Recommendation P.910*, April 2008.

[25] T. Hoßfeld, T. Zinner, R. Schatz, M. Seufert, and P. Tran-Gia, "Transport Protocol Influences on YouTube QoE ," Tech. Rep. 482, University of Würzburg, July 2011.

[26] T. Hoßfeld, R. Schatz, and S. Egger, "SOS: The MOS is not enough!," in *QoMEX 2011*, (Mechelen, Belgium), Sept. 2011.

[27] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "An Evaluation of QoE in Cloud Gaming Based on Subjective Tests," in *Workshop on Future Internet and Next Generation Networks*, (Seoul, Korea), June 2011.

[28] J. C. Platt, "Using Analytic QP and Sparseness to Speed Training of Support Vector Machines," in *Conference on Advances in Neural Information Processing Systems 11*, vol. 11, (Dever, USA), Nov. 1998.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.

[30] M. Fiedler, T. Hoßfeld, and P. Tran-Gia, "A Generic Quantitative Relationship between Quality of Experience and Quality of Service," *IEEE Network Special Issue on Improving QoE for Network Services*, vol. 24, June 2010.

[31] F. De Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro, and T. Ebrahimi, "Subjective assessment of H.264/AVC video sequences transmitted over a noisy channel," in *Proceedings of the First International Workshop on Quality of Multimedia Experience (QoMEX 2009)*, 2009.

[32] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling tcp throughput: a simple model and its empirical validation," *SIGCOMM Comput. Commun. Rev.*, vol. 28, pp. 303–314, October 1998.

[33] T. Hoßfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, and P. 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), Dec. 2011.