Network-Aware Video Level Encoding for Uplink Adaptive HTTP Streaming

Christian Lottermann[1,2], Serhan Gül[1], Damien Schroeder[2], and Eckehard Steinbach[2]
[1] BMW Group, Munich, Germany {christian.lottermann@bmw.de, serhan.guel@bmw.de}
[2] Chair of Media Technology, TUM, Munich, Germany {damien.schroeder@tum.de, eckehard.steinbach@tum.de}

***Abstract*—We study the uplink delivery of live video using adaptive HTTP streaming (AHS). In AHS, the process of simultaneously creating video levels at different rates is computationally demanding and quickly exceeds the computational capacity of mobile devices. As a remedy, we propose a network-aware video level selection approach which reduces the number of levels that need to be encoded. To this end, we develop an algorithm which selects a reduced set of video levels from a static pre-defined set based on TCP uplink throughput information. More specifically, during session start-up and after inter-RAN handovers, TCP uplink throughput information from a remote database is used while otherwise, actual TCP uplink throughput measurements are performed. We test the proposed approach in an automotive scenario to upstream the video of a vehicle's front-facing camera to a remote video portal. Our results show that our proposed network-aware video level selection approach leads to a significant reduction of the number of video levels that need to be encoded. At the same time, a similar quality of experience is achieved in terms of mean subjective quality of the delivered video segments, interrupted playback duration due to stalling events, and number of quality switches when compared to an implementation which considers the static full set of video levels.**

I. Introduction

Advanced capabilities of modern consumer electronic devices and the enhanced capacities of radio access networks (RANs) have led to high popularity of mobile video applications which are nowadays responsible for roughly half of the internet traffic and according to Cisco this trend will increase up to 70% in 2017 [1]. A significant proportion of the generated video content is upstreamed from mobile devices, such as smart phones, tablet PCs, or connected vehicles to video portals or transmitted directly to a user. The stream of a front-facing camera of a vehicle, for example, can be used in live streetview [2] or remote traffic monitoring systems [3].

Different streaming systems have been proposed. RTP/UDP-based systems are able to support intra-session adaptation using specialized streaming servers, but are often blocked by firewalls. On the contrary, HTTP/TCP based streaming systems do not suffer from firewall filters since almost all network nodes in the internet are configured to support HTTP traffic. Traditional HTTP based progressive download systems are widely used, however, lack intra-session rate adaptation which might lead to stalling under congested network conditions. Adaptive HTTP streaming (AHS) systems, such as the recently standardized dynamic adaptive streaming over HTTP (DASH) [4], support dynamic rate adaptation and are usually deployed in content delivery networks (CDNs) to downstream live and on-demand video content. At the source side, videos are encoded at different target video rates (video levels), divided into segments of a fixed duration and stored on a standard web server. Depending on the settings, usually a set of 10-15 video levels is generated. As discussed in [5], the levels need to be carefully selected considering the deployment properties of the streaming system. The adaptation of the video stream is controlled at the receiver side, where an AHS client adaptively requests segments at a rate that matches the current network performance. For example, the adaptation algorithm proposed in [6] considers the segment fetch time and the algorithms proposed in [7] and [8] additionally consider the buffer fullness for the video level selection.

Video streaming to and from mobile devices is especially challenging due to the time-varying network performance and inter-RAN handovers that might lead to stalling. While AHS systems are able to deal with changing network conditions using intra-session adaptation, they have not been considered for uplink live streaming from mobile devices so far since the process of creating the different video levels is computationally demanding. The computational resources of mobile devices are limited and hence such devices might not be able to create the same number of video levels in real-time as CDN systems. Nevertheless, due to its favorable deployment and transport characteristics, AHS offers major advantages compared to other systems for uplink streaming. One option to reduce the computational demand is to use scalable video coding with AHS [9], but it is still not clear whether the scalable extensions of modern codecs, such as H.264/AVC and H.265/HEVC, will find broad acceptance in the market.

In this paper, we propose a network-aware video level selection approach for uplink AHS which allows us to reduce the number of video levels to be encoded at the mobile device side for uplink AHS. To this end, we propose an algorithm that creates a reduced set of video levels based on two sources of TCP uplink throughput information. During the start-up phase and after inter-RAN handovers, TCP uplink throughput information for the current position and the connected RAN of the mobile device is requested from a remote network performance database. Otherwise, TCP uplink throughput measurements are performed and this information is used for the video level selection. We apply and test the algorithm in a mobile streaming scenario, where the video of a vehicle's front-facing camera is upstreamed to a video portal while using heterogeneous RANs. In our evaluation we use TCP network performance traces from measurements in real HSPA and LTE networks. Furthermore, we develop a perceptual rate-distortion model to select the video levels in a perceptual quality-aware manner. We define four key performance indicators (KPIs) to determine the performance of our approach in a streaming session: (i) the duration of interrupted playback due to stalling events, (ii) the number of quality switches, (iii) the mean bit rate of the transmitted segments, and (iv) the mean subjective quality of all transmitted video segments based on our developed perceptual rate distortion model. Based on the defined KPIs, we assess the performance of our proposed approach using three AHS standard adaptation algorithms [6, 7, 8] on the receiver side. Our proposed video level selection approach shows a similar performance as a reference implementation, which uses the full static set of video levels, while reducing the number of video levels by 83% and thus the encoding complexity significantly.

Uplink streaming has been considered in the context of peer-to-peer streaming systems in [10]. To the best of our knowledge, this is the first paper that considers AHS for uplink streaming from mobile devices while reducing the number of encoded video levels based on TCP uplink performance information. The main advantage of our approach is that the mechanisms and deployment characteristics of HTTP/TCP streaming can be used while achieving low computational demands for the AHS encoder.

The rest of the paper is organized as follows. Section II describes the system model for AHS uplink streaming. Section III presents our network-aware algorithm which generates a reduced set of video levels. We apply our approach in a vehicular environment in Section IV and determine its performance in Section V. Finally, Section VI concludes this work.

II. System model

We consider a live video delivery scenario where a video is streamed from a mobile device to a video portal using AHS (Fig. 1). An AHS enabled encoder is set up at the mobile device which splits the video into segments of length τ, encodes the segments at different bit rates, and stores the corresponding bitstreams on a locally installed web server. All segments are referenced in an index file, referred to as Media Presentation Description (MPD) in DASH, which is used to inform the client about the available bitrates. Furthermore, we consider a static set $\mathbf{V}$, which consists of M video levels at different pre-defined bit rates. The AHS receiver entity is operating on a video portal, which adaptively requests segments from the mobile device using a standard AHS adaptation algorithm. The video portal can act as an intermediate node for both live consumption and storage of the video contents for on-demand retrieval. Alternatively, a client (e.g., the owner of the vehicle) could directly access the video remotely. Between the mobile device and the video portal we assume cascaded networks which consist of the uplink RAN of the mobile device and a wide area network (WAN) infrastructure. We consider the uplink RAN of the video source side as the network performance bottleneck.

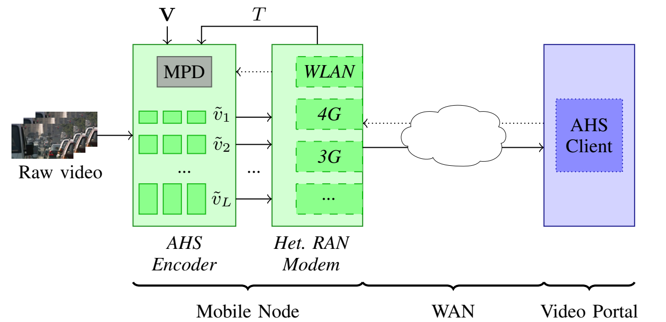


Fig. 1: System model for the TCP throughput-based adaptation of the video levels.

In the present work, we focus on the determination of a reduced set of target video levels, which contains a lower number of video levels compared to the static set, to decrease the computational demand for the parallel encoding of the video levels on the mobile device. More specifically, we introduce an algorithm to dynamically select a reduced set of video levels ($\tilde{\mathbf{V}}$) by filtering the static full video level set ($\mathbf{V}$) according to TCP uplink performance information. We assume that the RAN modem of the mobile device provides information about the measured uplink TCP throughput (T_{Meas}) for a window length of N seconds. At the session start-up and after an inter-RAN handover, typically reliable measurements about the TCP uplink throughput are difficult to obtain at the mobile device. Hence, we assume that geo-referenced TCP uplink throughput information (T_{DB}) is provided for the current position and connected RAN of the mobile device at a remote database that contains information about the previous TCP transmissions.

We assume that the AHS client at the video portal reuse a standard AHS adaptation algorithm which (i) downloads the video segments in chronological order, (ii) requests only one video level per download, and (iii) downloads the video segments in a non-preemptive manner, i.e., the download of the current segment must be completed before the start of the next segment. Furthermore, we assume that the MPD file at the client is updated once the adaptation of the reduced video level set is invoked.

III. Network-aware video level selection

The goal of the algorithm is to dynamically select a reduced set of video levels $\tilde{\mathbf{V}}$ from the static set $\mathbf{V}$ based on TCP uplink throughput information.

The algorithm's pseudocode is provided in Algorithm 1. We assume that the algorithm is invoked at time t immediately after the request of a segment when one of the two following cases occurs: (i) a novel streaming session is started or an inter-RAN handover between two RANs occurs, (ii) a window of N seconds is completed.

The algorithm takes the following input arguments: (i)$\mathbf{V} = \{v_1, v_2, \cdots, v_M\}$: the static full set of pre-defined video levels, (ii) L: number of levels in the reduced set of video levels, (iii) h: flag which is set to 1 after start-up and a inter-RAN handover and 0 otherwise, (v) T_{Meas}: measured TCP uplink

Algorithm 1: NETWORK-AWARE VIDEO LEVEL SELECTION

Input: $\mathbf{V} = \{v_1, v_2, \cdots, v_M\}$: static set of video levels
L: number of levels in the reduced set
h: flag which is set to 1 at startup and handover
T_{Meas}: Measured TCP uplink throughput
T_{DB}: TCP uplink throughput from remote database

Output: $\tilde{\mathbf{V}} = \{\tilde{v}_1, \tilde{v}_2, \cdots, \tilde{v}_L\}$: reduced set of video levels

1 **if** $h = 1$ **then**
2 $\quad T = T_{DB}$
3 **else**
4 $\quad T = T_{Meas}$
5 $v_n = \underset{\{v_i, 1 \leq i \leq M\}}{\arg\min} \|v_i - T\|$
6 **if** $n < \lfloor L/2 \rfloor + 1$ **then**
7 $\quad \tilde{\mathbf{V}} \leftarrow \{v_1, v_2, \cdots, v_L\}$
8 **else if** $n > M - \lceil L/2 - 1 \rceil$ **then**
9 $\quad \tilde{\mathbf{V}} \leftarrow \{v_{M-L+1}, v_{M-L+2}, \cdots, v_M\}$
10 **else**
11 $\quad$ **if** L *is odd* **then**
12 $\quad\quad \tilde{\mathbf{V}} \leftarrow \{v_{n-\lfloor L/2 \rfloor}, v_{n-\lfloor L/2 \rfloor+1}, \cdots, v_{n+\lfloor L/2 \rfloor}\}$
13 $\quad$ **else**
14 $\quad\quad \tilde{\mathbf{V}} \leftarrow \{v_{n-\frac{L}{2}}, v_{n-\frac{L}{2}+1}, \cdots, v_{n+\frac{L}{2}-1}\}$
15 **return** $\tilde{\mathbf{V}}$

throughput, (v) T_{DB}: TCP uplink performance information obtained from a remote database. The algorithm has one output argument, $\tilde{\mathbf{V}} = \{\tilde{v}_1, \tilde{v}_2, \cdots, \tilde{v}_L\}$: the reduced set of video levels.

Once the algorithm is invoked, the TCP uplink throughput information is determined (line 1-4). After the start-up and after an inter-RAN handover, h is set to 1 and thus, T is set to the TCP uplink throughput information T_{DB} of the current position and connected RAN of the mobile device requested from a remote database. At all other times, h is set to 0. Hence, the algorithm sets T to the measured TCP uplink throughput.

In the next step, the video levels of the reduced set $\tilde{\mathbf{V}}$ are determined according to T. First, the video level v_n is selected as the center element of $\tilde{\mathbf{V}}$, that offers the smallest absolute difference to T among the video levels in $\mathbf{V}$ (line 5). The algorithm selects the remaining video levels by choosing $L-1$ video levels around v_n (line 6-14). If L is an odd number, a symmetric selection is conducted (line 12). Otherwise, the algorithm takes a conservative approach and favors video levels with smaller rates compared to v_n (line 14). Depending on L, if not enough video levels are below or above v_n, the algorithm constructs $\tilde{\mathbf{V}}$ asymmetrically around v_n (lines 6-9).

IV. APPLICATION IN AUTOMOTIVE ENVIRONMENTS

We apply our proposed video level subset selection approach in an automotive environment where the video of a front-facing camera of a vehicle is upstreamed to a video portal using AHS. The AHS enabled encoder is installed on an electronic control unit (ECU) in a vehicle which supports connectivity to cellular networks (HSPA and LTE). The AHS client is installed at the video portal. This scenario is especially interesting since the network performance of moving vehicles fluctuates significantly over time due to fast-changing wireless channels and inter-RAN handovers. Besides that, the vehicle's ECUs offer limited computational capacity that support the parallel encoding of only a small number of video levels.

In the following, we describe the experimental setup for our performance assessment. We first describe TCP uplink throughput measurements from a vehicle to a server in the internet conducted in real HSPA and LTE networks. Next, we develop a perceptual rate-distortion model based on a subjective test to determine the expected subjective quality for video segments encoded at a certain bit rate. Finally, based on the TCP network performance and the perceptual rate-distortion model, we select the M video levels for the static video level set $\mathbf{V}$.

A. Network performance modeling

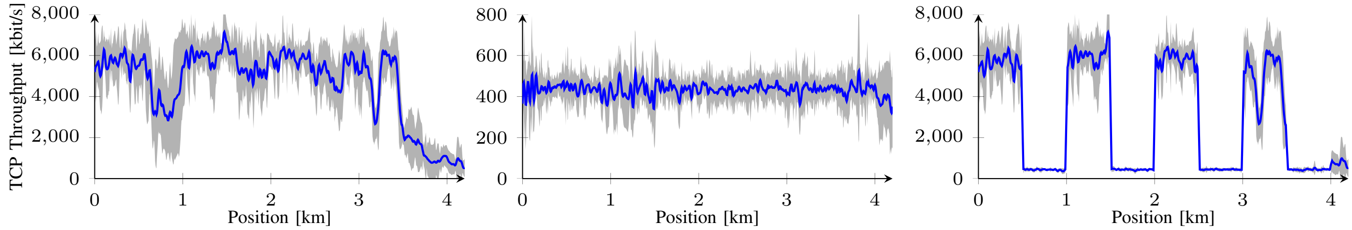
To model the TCP uplink performance of the RANs in an automotive environment, we conducted TCP uplink throughput measurements of a single TCP flow in an urban environment while driving. We recorded TCP uplink performance traces with a server in the internet using *iperf* [11] along a 4.3 km long track in the urban area of Munich with a ZTE MF821 data modem and a rooftop antenna system [12]. We performed nine repeated measurements in the network of Telefónica for both, HSPA and LTE networks. The measurements have an average duration of 504 s. All traces were recorded at an average velocity of 30 km/h.

We recorded data points at a frequency of 1 Hz containing uplink TCP throughput measurements, geographical position (latitude and longitude), velocity of the vehicle, and timestamp. Fig. 2a and 2b show the mean and standard deviation of the measured uplink TCP throughput over all conducted measurements for HSPA and LTE networks, respectively. We observe that with LTE, the mean uplink TCP throughput fluctuates significantly around a mean of roughly 5000 kbit/s (except for the significant drop in the last 500 m) whereas with HSPA, the TCP throughput is lower but also more stable with a mean of approximately 450 kbit/s. For further analysis, we create an artificial scenario where the vehicle performs handovers between HSPA and LTE networks. Therefore, we introduce periodic handovers that occur every 60 s and piecewise combine our geo-referenced LTE and HSPA traces obtained from respective uplink TCP throughput measurements, cf. Fig. 2c.

We calculate the mean over all traces for each RAN and position to create the TCP uplink performance information in the geo-referenced database.

B. Perceptual rate-distortion model

To quantify the subjective quality for the different video levels, we create rate-distortion curves using the mean opinion score (MOS) scale defined in [13], which ranges from of 1.0

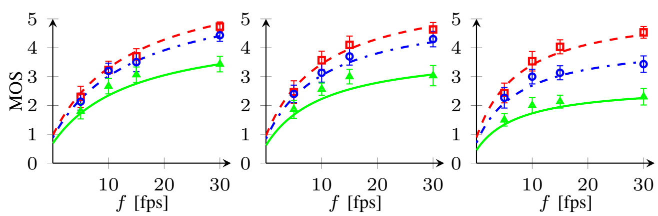


(a) LTE (b) HSPA (c) Handover

Fig. 2: Mean (———) and standard deviation (▇▇) of the measured uplink TCP throughput over nine traces.

(a) Video 1 (b) Video 2 (c) Video 3

Fig. 3: Example frames of the selected videos.



(a) Video 1 (b) Video 2 (c) Video 3

Fig. 4: Calculated MOS values based on trained STVQM [14]: 34 dB (———), 38 dB (· - · -), 42 dB (- - -); MOS obtained from subjective test: 34 dB (▲), 38 dB (○), 42 dB (□) with the 95% CI.

(worst) to 5.0 (best). For our investigation, we select three 10 s long videos from road scenes which are recorded at a resolution of 1280x720 and 30 frames per second using a prototypical front-facing camera. Example frames of the selected videos are displayed in Fig. 3.

We use the spatio-temporal video quality metric (STVQM) proposed in [14] to model the subjective quality for different spatial quality impairments and temporal resolutions. To train the model parameters we first conduct a test to assess the subjective quality based on SAMVIQ [15] for different spatio-temporal quality impairments. For each uncompressed source video (SRC) we create 12 processed video sequences at four different frame rates ($f \in \{30, 15, 10, 5\}$ fps). For each frame rate we encode the video at three different PSNR levels (42 dB, 38 dB, 34 dB) in H.264/AVC main profile at a constant quantization parameter using x264 [16]. We use the screening procedure proposed in [15] to exclude outliers from the ratings. After the screening procedure, 15 out of 17 votes are verified to be valid. Fig. 4 shows the mean MOS values of the subjective test results and the calculated MOS values of STVQM for the videos for different PSNR levels.

To produce the rate-MOS curves for the videos, we create sequences by encoding each SRC with a constant quantization parameter ($q \in \{10, \ldots, 50\}$) at different frame rates ($f \in \{30, 15, 10, 5\}$ fps) and an IPPP... GoP structure. We use different GoP lengths for each frame rate, such that one GoP has a length of $\tau = 2$ s, which is identical to the segment length. For the encoded video sequences we determine the

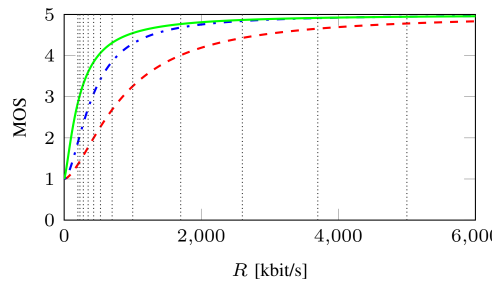


Fig. 5: Rate-MOS curves of video 1 (- - -), video 2 (· - · -), and video 3 (———); video levels of $\mathbf{V}$ (·········).

MOS values using the trained STVQM with the selected frame rates and the measured PSNR values. For each video, we perform an exhaustive search to select the encoded video sequences that maximize the MOS for given rate constraints and use a logistic fitting to finally generate the rate-MOS curves that can be used to create an arbitrary set of points for each video. In Fig. 5 the rate-MOS curves with the defined video levels of the three considered videos are displayed.

C. AHS full set video level selection

The selection of the set of video levels has a significant impact on the user satisfaction. We follow the guidelines of [5] to select the video levels in a perceptual quality-aware manner.

For the number of video levels in our full video level set $\mathbf{V}$, we employ the typical CDN settings [5] and set $M = 12$. To determine the distribution of video levels in $\mathbf{V}$, we consider two factors. First, we determine the lowest and highest video levels in $\mathbf{V}$ such that the uplink throughput range observed during our measurements in HSPA and LTE networks is covered. Secondly, we put emphasis on lower bit rates ($R \leq 1000$ kbit/s). Encoding more video levels at lower rates is valuable since the gains in terms of subjective quality are large in this range (Fig. 5). This means that a small increase in throughput results in a significant subjective quality gain. Conversely, the subjective quality increase becomes smaller and smaller as the bit rate grows.

Considering these two factors and the typical CDN settings, we construct the full video level set as follows (in kbit/s), which is illustrated in Fig. 5 as vertical lines:

$$\mathbf{V} = \{200, 230, 280, 350, 430, 530, 700, 1000, 1700, 2600, \\ 3700, 5000\}$$

V. Performance Evaluation

We implement a simulation framework in MATLAB to evaluate the performance of the network-aware video level

TABLE I: Performance overview of the *Reference* and *Proposed* implementations for $L = 2$, $N = 10\,\text{s}$, mean over nine traces each.

		$\overline{\sigma}$ [s]		$\overline{\nu}$		$\overline{\mu_r}$ [kbit/s]		$\overline{\mu_m}$ [MOS] Video 1		Video 2		Video 3	
		Ref.	*Prop.*	*Ref.*	*Prop.*	*Ref.*	*Prop.*	*Ref.*	*Prop.*	*Ref.*	*Prop.*	*Ref.*	*Prop.*
LTE	Liu	0.0	0.0	37	27	4365.2	4203.5	4.58	4.52	4.84	4.81	4.88	4.86
	Miller	0.0	0.0	19	16	4321.9	3544.0	4.54	4.41	4.80	4.77	4.85	4.84
	Tian	0.0	0.0	13	21	4383.6	3932.3	4.47	4.45	4.73	4.78	4.81	4.85
Handover	Liu	0.0	6.4	49	22	2804.5	2912.7	3.57	3.62	4.14	4.17	4.46	4.48
	Miller	4.9	6.2	47	20	2190.4	1928.8	3.21	3.27	3.65	3.95	4.07	4.35
	Tian	0.8	0.4	26	25	2775.7	2632.7	3.44	3.54	3.95	4.11	4.33	4.45

TABLE II: Feasible (N, L_{min})-pairs for Algorithm 1.

N [s]	10	20	40	60
L_{min}	2	4	4	5

subset selection described in Section IV. For our evaluation we consider two different implementations at the encoder side: (i) a reference implementation (referred to as *Reference*) where the AHS enabled encoder produces the full set of M video levels with no network-aware adaptations, (ii) an implementation where our proposed network-aware video level selection approach (referred to as *Proposed*) is applied. At the AHS client side, we consider three different standard AHS adaptation algorithms: [6] (Liu), [7] (Miller), and [8] (Tian). For consistency, we set the parameters of all three algorithms to offer a mean buffer fullness of 30 s in the LTE scenario using the *Reference* implementation, which is feasible for most live streaming scenarios. For the performance analysis we introduce four KPIs determined at the AHS receiver side (σ, ν, μ_r, μ_m). We define σ as the total duration of stalling events in a streaming session when the buffer is empty and the playback of the video stream is interrupted. A large interrupted playback duration leads to a low user satisfaction. The *number of video level switches* (ν) in a session indicates the amount of quality changes of a streaming session. A lower number of switches during a session leads to a higher overall user satisfaction [17]. We calculate ν as

$$\nu = \sum_{i=0}^{I} a(v_i), \qquad a(v_i) = \begin{cases} 0 & v_{i-1} = v_i \\ 1 & v_{i-1} \neq v_i,\, i = 0 \end{cases}$$

where i is the index of the fetched segment, v_i the video rate of the ith fetched segment, and I the number of fetched segments in the session. The *mean video rate* μ_r defines the mean bit rate of the transmitted video levels for the streaming session and is determined by

$$\mu_r = \frac{1}{I} \cdot \sum_{i=1}^{I} v_i.$$

Lastly, we calculate the mean subjective quality of the transmitted video segments of a streaming session (μ_m) by

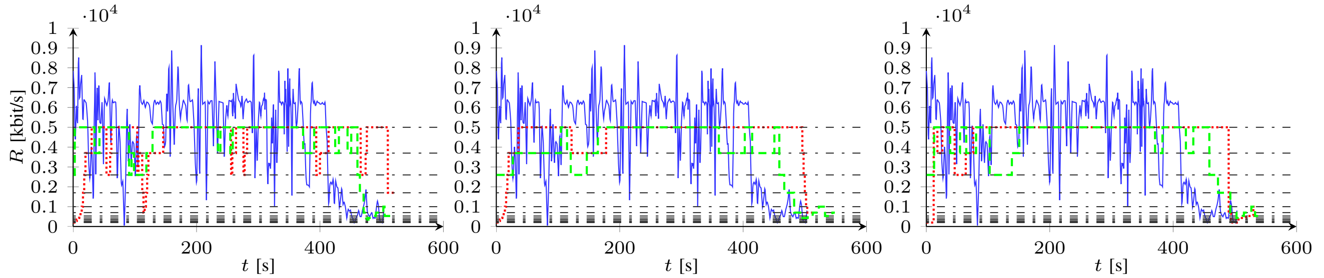
$$\mu_m = \frac{1}{I} \cdot \sum_{i=1}^{I} m(v_i),$$

where $m(v_i)$ is the MOS value for v_i based on the perceptual rate-distortion model of Section IV-B.

We first investigate the influence of the encoder side adaptation time N of our *Proposed* implementation and determine the minimum number of video levels L_{min} that is required to achieve the same perceptual quality as for the *Reference* implementation. Hence, for each $N \in \{10, 20, 40, 60\}$ s, we determine L_{min} such that the perceptual quality, i.e., the mean subjective quality of the transmitted video segments, interrupted playback time due to stalling events, and number of video level switches is equal to or better than the *Reference* implementation. For this investigation, we use the TCP uplink throughput traces of the LTE measurements described in Section IV-A and employ Liu's adaptation algorithm [6] on the client side. Table II displays the obtained (N, L_{min})-pairs. We determine that L_{min} increases for larger N values since the responsiveness of our *Proposed* implementation to the throughput fluctuations decreases as N increases. For $N = 10\,\text{s}$ the lowest possible L_{min} value can be achieved ($L_{min} = 2$), which is an acceptable number of video levels that most modern vehicles equipped with cameras can encode in parallel.

Hence, we now determine the performance of our video level selection approach based on Algorithm 1 with the parameters ($N = 10\,\text{s}$, $L = 2$) and calculate the mean performance for all four KPIs ($\overline{\sigma}$, $\overline{\nu}$, $\overline{\mu_r}$, $\overline{\mu_m}$) over all TCP uplink throughput traces of the single RAN (LTE) and handover scenarios. The results of our investigation are displayed in Table I for all three client adaptation algorithms. As expected from our initial investigation, the performance of our *Proposed* implementation in terms of mean subjective quality and interrupted playback time due to stalling events is comparable to the performance of the *Reference* implementation for all client side adaptation algorithms and is below 1% of the overall streaming session duration. Moreover, for Liu's [6] and Miller's [7] adaptation algorithms, a decrease in the number of quality switches is observed. The main reason for this effect is that a small number of video levels leads to a decreased probability of quality switches.

Furthermore, we determine that the mean video rate obtained with our *Proposed* implementation is mostly lower than the mean video rate obtained with the *Reference* implementation. However, this does not necessarily lead to a lower mean subjective quality of the transmitted video segments over the streaming sessions since MOS is not linearly proportional to the video rate, cf. Fig. 5. Hence, the mean subjective quality depends highly on the transmitted video segments during the streaming process. Fig. 6 shows the transmitted video segments for an exemplary video streaming scenario using one LTE network trace for the *Reference* and our *Proposed* implementations, respectively. We determine that for our *Proposed*



(a) Liu [6]

(b) Miller [7]

(c) Tian [8]

Fig. 6: Exemplary LTE TCP uplink network throughput trace (——) with the video level decisions for the *Reference* (········) and *Proposed* (- - - -) implementations for different client adaptation algorithms ($N = 10\,\mathrm{s}$, $L = 2$); video levels of $\mathbf{V}$ (· - · -).

implementation, higher video levels are requested in the start-up phase, since Algorithm 1 already adapts the video level set at $t = 0$ based on the network statistics T_{DB}, which is not considered in the *Reference* implementation.

Hence, using our *Proposed* implementation with appropriate N and L settings, a similar overall perceptual quality compared to the *Reference* implementation can be achieved at a significantly lower number of video levels.

VI. Conclusion

In this paper, we propose to apply an AHS system to upstream videos from a mobile device with limited computational capacity. To this end, we develop a network-aware video level selection approach on the mobile device side to select a reduced set of video levels from a full static video level set according to TCP uplink throughput information. At the start-up and after an inter-RAN handover, the TCP uplink throughput for the current position and connected RAN of the mobile device is requested from a database whereas otherwise, measured TCP uplink throughput information is used. We apply the network-aware video level selection approach in an automotive environment where the video of a vehicle's front facing camera is upstreamed to a video portal using two RANs (HSPA and LTE). We develop a perceptual rate-distortion model to select the rates of the video levels of the full static video levels set in a perceptual quality-aware manner. The results for three different AHS client adaptation algorithms show that the number of video levels can be reduced from the 12 pre-defined video levels all the way down to two dynamically selected video levels. At the same time, a similar performance with respect to the perceptual quality, i.e., mean subjective quality of the transmitted video segments over the streaming sessions, interrupted playback duration due to stalling events, and number of video level switches compared to a reference implementation can be achieved. Our proposed network-aware video level selection approach is well suited for uplink streaming of videos from mobile devices since it reduces the computational demands of the video level creation significantly while offering the advantages of HTTP/TCP based streaming. We expect that our proposed methodology can also be applied for other mobile devices with limited computational capacity equipped with cameras, such as smart phones and surveillance cameras.

VII. References

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2013-2018," 2014.

[2] P. Gomes, C. Olaverri-Monreal, and M. Ferreira, "Making Vehicles Transparent Through V2V Video Streaming," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 930–938, 2012.

[3] A. Vinel, E. Belyaev, B. Bellalta, and H. Hu, "Live video streaming in vehicular networks," in *Communication Technologies for Vehicles*, vol. 8435 of *Lecture Notes in Computer Science*, pp. 156–162. Springer International Publishing, 2014.

[4] T. Stockhammer, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *ACM MMSys*, 2011, pp. 133–144.

[5] L. Toni, R. Aparicio-Pardo, G. Simon, A. Blanc, and P. Frossard, "Optimal set of video representations in adaptive streaming," in *ACM MMSys*, 2014, ACM MMSys, pp. 271–282.

[6] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate Adaptation for Adaptive HTTP Streaming," in *ACM MMSys*, 2011, pp. 169–174.

[7] K. Miller, E. Quacchio, G. Gennari, and A Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *PV 2012*, May 2012.

[8] G. Tian and Y. Liu, "On Adaptive HTTP Streaming to Mobile Devices," in *PV 2013*, Klagenfurt, Austria, Dec 2013, pp. 1–8.

[9] C. Müller, D. Renzi, S. Lederer, S. Battista, and C. Timmerer, "Using scalable video coding for dynamic adaptive streaming over HTTP in mobile environments," in *EUSIPCO*, Aug 2012, pp. 2208–2212.

[10] B. Rainer, C. Timmerer, P. Kapahnke, and M. Klusch, "Real-time multimedia streaming in unstructured peer-to-peer networks," in *IEEE CCNC*, Jan 2014, pp. 1136–1137.

[11] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf: The TCP/UDP bandwidth measurement tool," Accessed January 23, 2015.

[12] L. Ekiz, A Thiel, O. Klemp, and C.F. Mecklenbrauker, "MIMO performance evaluation of automotive qualified LTE antennas," in *EUCAP*, April 2013, pp. 1412–1416.

[13] ITU-R Recommendation, "P-800: Methods for subjective determination of transmission quality," *ITU*, 1996.

[14] Y. Peng and E. Steinbach, "A novel full-reference video quality metric and its application to wireless video transmission," in *IEEE ICIP*, 2011, pp. 2517–2520.

[15] ITU-R Recommendation, "BT.910: Subjective video quality assessment methods for multimedia applications," *ITU*, 2008.

[16] VideoLAN, "x264 project," http://www.videolan.org/developers/x264.html, Accessed January 23, 2015.

[17] P. Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen, "Spatial flicker effect in video scaling," in *QoMEX*, Sept 2011, pp. 55–60.