Model for estimating QoE of Video delivered using HTTP Adaptive Streaming

Johan De Vriendt, Danny De Vleeschauwer, David Robinson
Bell Labs, Alcatel-Lucent, Antwerp (Belgium)

Abstract— **The end user QoE (quality of experience) of content delivered over a radio network is influenced by the radio conditions in the RAN (radio access network). This paper analyses various QoE models for video delivered over a radio network (e.g. LTE (long term evolution)) using HAS (HTTP adaptive streaming). All QoE models analyzed are based on the HAS profiles constructed by intercepting the HTTP get messages. The performance of all optimally trained models is compared and the sensitivity of the models to unseen profiles, content and devices is investigated**.

Keywords — video; Quality of Experience; HTTP adaptive streaming; Mean Opinion Score ; subjective testing

I. Introduction

More and more video traffic is consumed over the open Internet via fixed and mobile access networks [1]. This content is delivered over BE (best effort) networks and is consumed on various types of new devices (traditional televisions, personal computers, tablets, smartphones).

A popular technique to deliver video streams over a BE network that may be congested to clients that may reside behind firewalls, is HTTP (hyper text transfer protocol) adaptive streaming [2][3][4][5]. In such a technique the video is segmented in intervals of between 2 and 10 seconds long and each video segment (i.e., consecutive non-overlapping interval of video) is encoded in multiple quality versions, where a higher quality version requires a higher bit rate to be delivered. The bit strings associated with these encodings are referred to as chunks. The HAS client uses HTTP to request chunks – the first ones usually at a low bit rate to build up a play-out buffer. If chunks are (consistently) delivered in a time shorter than the interval length, and when the play-out buffer is large enough, the RDA (rate decision algorithm) in the client selects a higher bit rate for the next chunk, such that download time is about equal to the interval length, keeping the play-out buffer more or less steady. In that way the RDA continually senses the available throughput and adapts the video rate accordingly. Any mismatch between video rate and throughput, is absorbed by the play-out buffer. Assuming the RDA is working properly, and there is enough bandwidth to support the lowest quality level, no stalling events should occur.

Various aspects can impact the QoE (quality of experience) of the end users: the physical layer (e.g., SINR (signal to interference and noise ratio)), the data link layer (e.g., BER (bit error rate)), the IP layer (packet delay, i.e., mean and variation, and packet loss), the TCP layer (e.g., goodput) and the application layer (e.g., RDA). In this paper we rely on the fact that the video client sends request messages to the video server (HTTP get messages). Intercepting those messages it can be accurately predicted in which quality each segment is played out. It is not exactly known when these segments are played out and the video could stall, but since HTTP adaptive streaming was especially designed to avoid such stalls we do not take them into account here. Moreover, from other work [6] it is clear that even one stall degrades the QoE to an unacceptable level. Consequently, we solve the problem of how to assess QoE of an end user under the form of a prediction for the MOS (mean opinion score) an audience of users would give to a video played out continuously consisting of segments that may have different qualities. Several QoE models will be analyzed that predict the MOS based on the identified HAS profile.

The rest of this paper is organized as follows. In the next section we describe related work. Section III explains how the data upon which we tuned the model was acquired. In Section IV the models are introduced, of which the performance is assessed in section V. Finally Section VI draws the conclusions.

II. Related Work

QoE of video has received a lot of attention lately. Basically there are two types of models: models that work in the pixel domain and models that work on the bit stream.

In the first type of models the video needs to be decoded and processed and depending on how much of the original video sequence is needed in order to make a MOS prediction various classes can be distinguished: 1) full-reference models (in which the complete sequence is needed), 2) reduced-reference models (in which features calculated on the original video sequence are needed) and 3) no-reference model (which solely operate on the received sequence). Typical examples of a full-reference model are PSNR (peak signal to noise ratio) and SSIM (structural similarity). These models have in common with the models that described in this paper is that they typically calculate a quality metric for each image and need to translate this into a value for the sequence. This is similar to our model that has a quality indication associated with a segment and needs to provide the QoE of a sequence of chunks. In the literature typically averaging and pooling is used for this process. Averaging just takes the average, while pooling [10][11] only averages over the lowest ranked values. In this paper we use averaging but subtract a suitably scaled standard deviation.

The second type of models inspects the video flow and makes a prediction for the quality based on characteristics of this flow (without actually decoding the video). Typically the delay (i.e., its average and variation) and packet loss (see e.g., [7]) are used as network parameters and the end devices are queried with respect to the codec rate and the depth of the

978-3-901882-50-0 ©2013 IFIP

dejitter buffer that are used. This technique is only applicable to video transported over RTP/UDP (real-time transport protocol/user datagram protocol) and cannot directly be used in HTTP adaptive streaming because TCP (transport control protocol) that is used to transport the HTTP responses (to the HTTP get messages) retransmits lost packets, so that the video client sees reliable delivery. The delay - in fact the RTT (round trip time) - and packet loss do impact the throughput that the client sees and Padhye's formula [8] predicts the throughput when RTT and packet loss are given. However, if the throughput needs to be known, it is better to measure it directly by counting packets or inspecting the ACKs. But, even if the throughput is known, it is difficult to assess the quality of HTTP adaptive streaming, because different RDAs react differently to the same throughput profile [9]. So rather than measuring the throughput and trying to assess how the RDA would react to it, it is better to monitor what the RDA actually decides (by inspecting the HTTP get messages) and base the quality assessment on this information. This approach, which we follow in this paper, is novel, as far as we know.

III. Data Collection

The end user QoE of content delivered over a radio network is influenced by the radio conditions in the RAN (radio access network). This RAN is shared between the active users within the cell. For any one user, its share of the bandwidth depends on factors such as fading, SINR (signal to interference and noise ratio), RTT and the amount of other concurrent traffic (both upstream and down). We defined 84 radio scenarios. A scenario is a setting for the network conditions and is defined as a combination of the SINR profile (e.g. SINR reducing from 30 dB to 0 dB in steps of 1 dB every 2 seconds), the number of competing FTP sources, the RTT, the fading profile and the clip (see further).

To obtain the subjective quality of content, the conventional approach is to recruit a representative group of volunteers. These volunteers are asked to rate a selection of clips during one or more viewing sessions, while ensuring that the viewing conditions are carefully controlled. This approach is not feasible as we wanted to explore the impact of several network impairments.

However, an important property of HAS is that it is possible to playback a video session. This is because it is possible to identify the video levels of each chunk in that session. In particular, we can create a file by concatenating the chunks requested for segment 1, followed by the chunk requested for segment 2 and so on. The resulting file can be downloaded to a volunteer's device and played back locally. Hence we can use the 'wisdom of the crowds' and recruit many more viewers without the need for them to attend a viewing session in person.

We chose two clips of nominally 2 minutes long from Sintel www.sintel.org: the first one from the opening sequence panning over the mountainscape and the second one of high motion including a chase through a market. The content is encoded in 6 levels (128, 210, 350, 545, 876 and 1440 kbps) AVC and 64kbps stereo audio.

Figure 1 shows the set-up with which we emulated an LTE (long term evolution) network in which we could simulate typical network impairments.

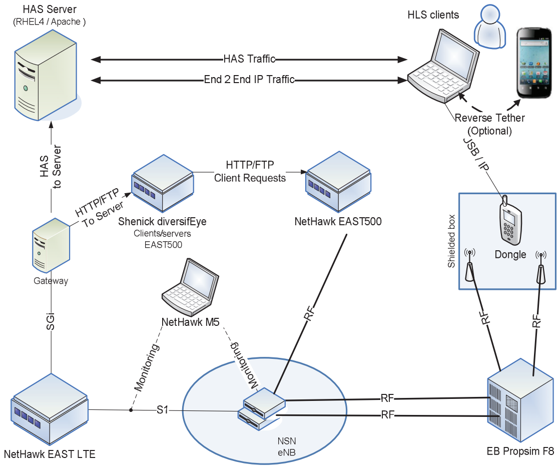

Figure 1: Lab environment

The lab setup consists of a HAS server connected to the HAS clients via an LTE network. We chose the Apple HLS (HTTP live streaming) [3] variant of HAS running on an Apple Mac mini connected to an LTE dongle. This dongle was in a shielded box along with the aerial. The Propsim F8 allowed the controlled introduction of various radio related impairments and the NetHawk East T500 allowed injecting background traffic to alter the congestion in the cell.

After configuring the Propsim F8 and the NetHawk East T500 corresponding to 1 of in total 84 scenarios we started the HLS client on the Mac mini. The corresponding HAS profile is found in the "Access Log" on the HAS Server. Each of the 84 scenarios was run on average 8 times and a representative HAS profile was identified for that scenario.

Using the HAS profile, a video file for each scenario was created. This was done by concatenating the HAS chunks starting with the chunk for quality level for segment 1, followed by the chunk for segment 2 and so on. From initial analysis, it is clear that some HAS profiles are similar so there was no need to test all scenarios. This allowed us to introduce some synthetic profiles – HAS profiles which did not show up when using the Apple HLS client in our test network but could occur with other RDAs.

The resulting 90 video files (38 for clip 1 and 52 for clip 2) were posted on a website and we recruited some 500 volunteers prepared to take part in a brief survey and rate 5 of the 90 clips. The volunteers all had to have access to either an iPhone or iPad. By averaging the scores associated with a video file viewed on a specific device (iPhone or iPad), we obtained the MOS M_{subj} for that video on the respective devices.

IV. Models

We refer to a HAS profile P as a sequence of numbers $(l_1, \ldots, l_k, \ldots, l_K)$ where $l_k \in \{1, 2, \ldots, L\}$ indicates that in segment k the video was played-out in quality l_k with L the number of quality levels (l=1 indicates the lowest quality and l=L the highest) and with K the length of the profile (with k=1 indicating the most recent segment). As explained in the previous paragraph we have 90 of such profiles with an associated MOS value per device type.

To each chunk, characterized by a pair (k,l), values $M_{k,l}$ and $S_{k,l}$ are associated that give an indication of the quality of that chunk. Examples of $M_{k,l}$ are the nominal bit rate, the average PSNR or SSIM averaged over all frames of that chunk or a MOS value for that chunk. Examples of $S_{k,l}$ are the standard deviation of the PSNR or SSIM of all frames of chunk k at quality level l.

We consider models that predict the MOS for profiles given the value M_{k,l_k} and S_{k,l_k} . In particular, the predicted MOS M_{pred} is given by:

$$M_{pred} = \alpha \cdot \mu - \beta \cdot \sigma - \gamma \cdot \phi + \delta$$

where α, β, γ and δ are (tunable) parameters and μ (average of quality info), σ (standard deviation of quality info) and ϕ (frequency of switches) are metrics associated with the profile:

$$\mu = \frac{\sum_{k=1}^{K} M_{k,l_k}}{K}$$

$$\sigma = \sqrt{\frac{\sum_{k=1}^{K} \left(M_{k,l_k} - \mu\right)^2 + \sum_{k=1}^{K} S_{k,l_k}^2}{K}}$$

$$\phi = \frac{\sum_{k=1}^{K-1} 1_{\{l_k \neq l_{k+1}\}}}{K-1}$$

We considered alternative ways to calculate this predicted MOS (using weight factors to calculate the moments or a percentile of the $M_{k,l}$ values), but since initial results have shown that the performance gain was marginal, we do not discuss these alternatives here further.

In order to tune the parameters α, β, γ and δ and possibly $M_{k,l}$ and $S_{k,l}$ we minimize the RMSE (root mean squared error) between what the model predicts and the measured subjective MOS values. In particular, let $M_{subj,n}$ be the MOS associated with the n-th in a set of N HAS profiles (for a specific device and clip), obtained as described in Section III. Then, we tune the parameters by minimizing the RMSE, or minimize:

$$\frac{\sum_{n=1}^{N} \left(M_{pred,n} - M_{subj,n}\right)^2}{N}$$

This sum can be taken over all profiles (over all devices and clips) or over a subset of profiles, e.g. per (clip, device) combination or over a training and test subset.

In particular, there are four (types of) models that we consider.

1. In the (nominal) bit rate model (labeled “BR” below), $M_{k,l}$ is independent of k and equal to the nominal bit rate associated with quality level l, while $S_{k,l}$=0.
2. In the PSNR, respectively SSIM, model (labeled “PSNR” and “SSIM” below) $M_{k,l}$ is, the average PSNR, respectively SSIM, and $S_{k,l}$ is the standard deviation of the PSNR, respectively SSIM, both taken over all frames of that chunk.
3. In the chunk-MOS model (labeled “CM” below) $M_{k,l}$ is independent of k and defined as the MOS associated with quality level l, while $S_{k,l}$=0. In this case we consider two variants:
   a. The $M_{k,l}$ (one value per quality level) is either provided or needs to be estimated as part of the RMSE minimization process (in which case it can easily be seen that α and δ are redundant parameters).
   b. The $M_{k,l}$ (one value per quality level) are assumed to be uniform spaced between a minimum and maximum.
4. In the quality level model (labeled “QL” below) $M_{k,l}$=l (independent of k) while $S_{k,l}$=0. It can be easily proven that this is equivalent to model 3.b.

V. Performance

A. Data set

For each of the 90 videos (corresponding with the 90 HAS profiles) mentioned in section III we obtained between 11 and 16 subjective scores using a tablet (iPad) and between 5 and 10 scores using a smartphone (iPhone). The number of scores is low, especially for the phone, and this limits the accuracy that can be obtained. The average (taken over the 90 profiles and two devices) of the standard deviation of the scores per profile is about σ_S=0.84. Taking into account the number of scores we obtained per profile we estimate the error in the subjective MOS ($\sigma_S/\sqrt{M}$, with M number of scores per profile) on average to be 0.24 for the tablet, and 0.32 for the phone. These numbers give an indication of the RMSE that can be obtained with any model, as the RMSE will be a combination of the error in the MOS model and the error in the MOSs from the subjective tests.

Figure 2 and Figure 3 show that the 90 identified profiles are well spread over the (μ, σ, ϕ) space. However, in the ϕ dimension the spread is relatively low, but this is consistent with the HAS profiles we captured during the lab tests. The profiles with a larger ϕ are part of the synthetic profiles that were added.

B. Metrics

Although the models described in section IV have been trained by minimizing the RMSE, we also considered other performance metrics, i.e., the PCC (Pearson (or linear) correlation coefficient) and the SROCC (Spearman rank order correlation coefficient).

C. *Parameter tuning*

The parameters of each of the models have been trained by minimizing the RMSE over the subset of profiles corresponding with one clip and one device. This allows us to compare the different models optimized for each (clip, device) pair.

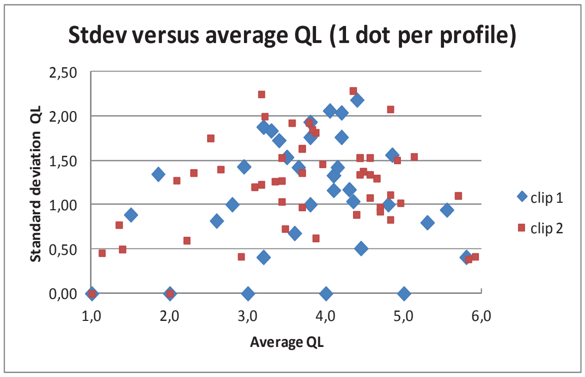

Figure 2: Profile description in the average QL (μ) and stdev QL (σ) space

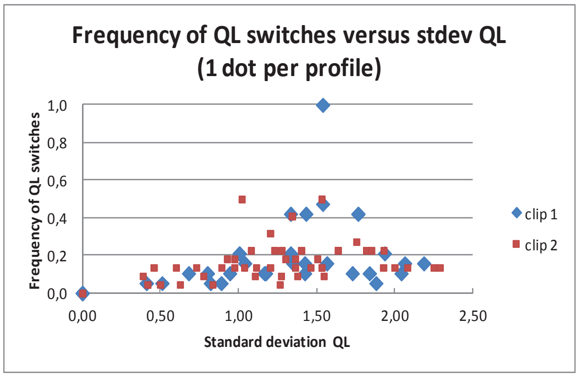

Figure 3: Profile description in the stdev QL (σ) and frequency of QL switches (ϕ) space

The results for the RMSE are shown in Figure 4. This figure shows that when the model parameters are optimized for (clip, device) pair the chunk-MOS model gives the best performance for the tablet for both clips and for the phone and clip1. The SSIM model gives the best performance for the phone and clip2. These conclusions are valid for all 3 metrics: RMSE, PCC, and SROCC. The SSIM model is always better than the PSNR model. The QL model is inferior to the chunk-MOS model (which is obvious, because as explained before the QL model is a special case of chunk-MOS model). The BR model gives the worst performance. This can be explained by the fact that the relation between bit rate and MOS cannot be accurately modeled by a linear relation. A modified BR model where $M_{k,l}$ values are a non-linear transformation of the bit rates will give better results.

When analyzing the obtained results we also observed that the impact of the profile metric ϕ (number of QL switches) is small, hence γcan set equal to 0. This is a general conclusion across all models, clip and devices. Consequently, we can conclude that the number of bit rate switches is not a key parameter to predict the MOS for the profile domain that we investigated as long as the standard deviation is taken into account. Investigating the space of profiles with a much large number of switches (profiles not seen frequently in the lab test) may result in a different conclusion.

The RMSE values for the chunk-MOS models are close to the estimated error in the subjective MOS values of the dataset. This gives an indication that the chunk-MOS model is quite good, and that the main contribution to the RMSE stems from the noisy $M_{subj,n}$. We also expect that with a dataset that has many more scores per video, the RMSE of the chunk-MOS model will be lower than the RMSE values we obtained here.

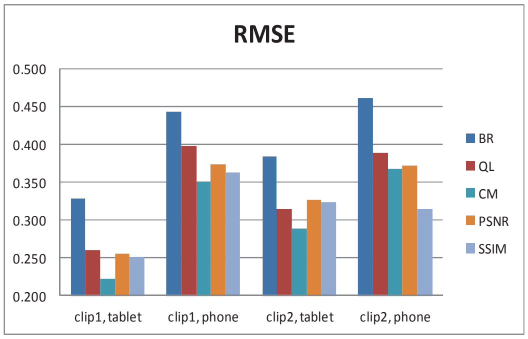

Figure 4: RMSE

D. *Sensitivity Analysis*

In the above section we have optimized the model by training the model over all profiles of the (clip, device) pair. It is also important to understand the sensitivity of the model when predicting the MOS value for unseen profiles; and for unseen content and devices. Therefore, 2 types of sensitivity analysis have been done. The first sensitivity analysis separates the profiles in a training and a test set. The second one analyses the sensitivity of the model parameters when using it for other content and/or devices.

1) *Sensitivity Analysis - Test and training set*

First, we will investigate the performance of the chunk-MOS model by splitting the set of profiles into a training set and a test set. The model parameter tuning was done based on the training set data, and the performance was then calculated for both the training set and the test set (all profiles except those of the training set). The size of the training set was varied between 11 and 35 (for clip1) and 11 and 48 (for clip2). For each size (T) of a training set, 30 different training sets were randomly picked from the complete set and the rest of profiles were used as test set. The performance was averaged over the 30 instances. This analysis was done for all (clip, device) combinations and qualitatively very similar results are obtained. In this paper we will illustrate the results by showing those for (clip1, tablet).

Figure 5 shows the average RMSE (averaged over the 30 instances) as a function of the training set size for the chunk-MOS model. As expected, the RMSE for the training set tends to increase as T increases. Indeed, the lower T, the better the model can be fit to the training set. On the other hand, the RMSE of the test set tends to decrease for increasing T, as for increasing T, the model is tuned for a larger set of profiles and will be valid more broadly. Therefore, the MOS value of previously unseen profiles can be more accurately estimated. From the figure it can be concluded that at least 20 profiles are needed in the training set to come close to the saturation level in the RMSE but even then a larger training set will further improve the MOS model, resulting in lower RMSE for the test set. Furthermore, it indicates that the error of the model will be higher than what you could conclude from the results in section V.C. Based on the dataset we have, the RMSE for (clip1, tablet) will be in the range of 0.29, while when optimizing the parameters for all profiles the RMSE was 0.225.

Similar qualitative conclusions are valid for the PCC and SROCC. PCC and SROCC for the training set decrease with increasing *T*, while those metrics increase for the test set with increasing *T*.

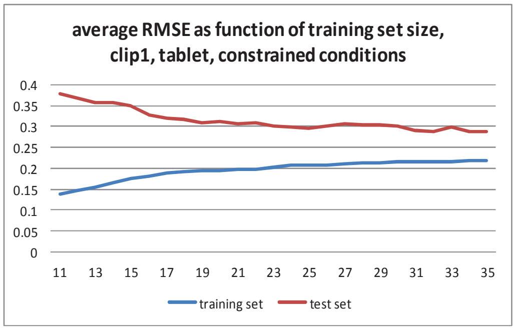

Figure 5: RMSE as function of the training set size.

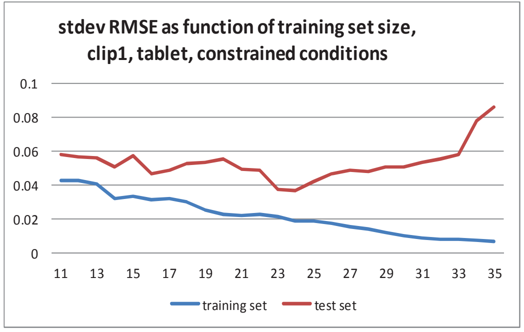

Figure 6: stdev RMSE as a function of the training set size.

Interesting is also to see how the parameters change over the 30 different instances of the training set. The average value of the parameters fluctuate little as a function of *T*, but this is not the case for the stdev (standard deviation) of the parameter values. It shows that the parameters become more stable for larger *T*.

The standard deviation of the RMSE (Figure **6**) of the training set decreases with increasing *T*. This is due to two effects: model parameters become more stable as they are tuned using more profiles, and the RMSE is calculated over a larger number of (*T*) profiles. The standard deviation of the RMSE of the test set initially decreases as the model parameters become more stable; but increases again for larger *T* due to the smaller size of the test set.

2) Sensitivity analysis across content and device

In order to analyze the sensitivity of the models for unseen content and devices, we evaluated the performance of the models when using the parameters (α, β, γ and δ) optimized for one (clip, device) pair for the other (clip, device) pairs (for which they are not optimal). The results of this analysis are shown in Table 1 and Table 2.

For the chunk-MOS model, the impact is quite moderate, with an increase of the RMSE of at most 0.125. Choosing a good compromise for the parameter values such as (β=0.26, γ=0.19), i.e., averaging over the 4 (β, γ) pairs, lowers this increase in RMSE to 0.034. And even when fixing γ=0 (and adjusting β=0.32) this increase in RMSE is limited to 0.032. In general we found that for all cases, the chunk-MOS model has a very broad minimum around the optimal (β, γ), and hence, is not very sensitive to these parameters. The reason for this is that a lot of the content and device specific information is present in the $M_{k,l}$ values (which do depend on clip and device type).

Table 1: Sensitivity across clip device of CM model

	β, γ optimal for				β=0.32, γ=0	β=0.26, γ=0.19
RMSE for	clip1		clip2			
	tablet	phone	tablet	phone		
clip1, tablet	0.223	0.296	0.272	0.286	0.255	0.242
clip1, phone	0.401	0.350	0.411	0.428	0.379	0.384
clip2, tablet	0.342	0.413	0.288	0.291	0.303	0.306
clip2, phone	0.428	0.492	0.370	0.368	0.390	0.394

Table 2: Sensitivity across clip, device of QL, PSNR and SSIM model (γ=0)

QL model	optimized for						
RMSE	clip1, tablet	clip1,phone	clip2, tablet	clip2, phone	tablet	phone	overall
clip1, tablet	0.260	0.542	0.318	0.406	0.303	0.467	0.286
clip1, phone	0.627	0.398	0.769	0.454	0.749	0.413	0.541
clip2, tablet	0.364	0.731	0.319	0.560	0.325	0.641	0.429
clip2, phone	0.510	0.452	0.628	0.398	0.614	0.411	0.446
PSNR model	optimized for						
RMSE	clip1, tablet	clip1, phone	clip2, tablet	clip2, phone	tablet	phone	overall
clip1, tablet	0.256	0.546	1.201	1.601	0.435	0.798	0.602
clip1, phone	0.609	0.373	0.814	1.212	0.457	0.477	0.409
clip2, tablet	1.100	0.517	0.327	0.579	0.492	0.553	0.461
clip2,phone	1.569	0.933	0.605	0.371	0.820	0.518	0.640
SSIM model	optimized for						
RMSE	clip1, tablet	clip1, phone	clip2, tablet	clip2, phone	tablet	phone	overall
clip1, tablet	0.251	0.539	1.220	2.205	0.404	0.711	0.522
clip1, phone	0.598	0.362	0.805	1.753	0.589	0.467	0.470
clip2, tablet	2.173	1.605	0.323	0.583	0.390	0.540	0.412
clip2,phone	2.633	2.061	0.578	0.314	0.702	0.445	0.541

The QL model is more sensitive to the model parameters (α, β and δ; γ=0 in Table 2), resulting in larger deltas in the RMSE when using model parameters that were tuned for another content and/or device.

The PSNR and SSIM model are both very sensitive to the model parameters. As an example, using the optimal SSIM model parameters for (clip1, tablet) for the (clip2, tablet) case (for which it was not tuned) yields an RMSE of larger than 2. In fact the predictions even do not stay within the [1, 5] range. When optimizing the model parameters over the complete set of profiles, i.e. all (clip, device) pairs, the RMSE deteriorates up to 0.35 for PSNR and up to 0.27 for SSIM. The reason for this bad performance of the PSNR and SSIM models while they had sometimes the best performance for the optimal parameter set is that these measures (PSNR, SSIM) are no good indicators for absolute quality. PSNR and SSIM give a good assessment of relative quality of different encodings of the same video, but are not good in assessing the absolute quality of different videos. This is illustrated in Figure 7. The $M_{subj,n}$ values for both clips vary roughly between 1.5 and 4.5, while the PSNR values of the different encodings for clip1 are almost consistently larger than the ones for clip2, and the same is valid for SSIM. This means that using the same PSNR or SSIM model parameters, the MOS values for clip1 will be predicted consistently larger than for clip2, while this is not the case.

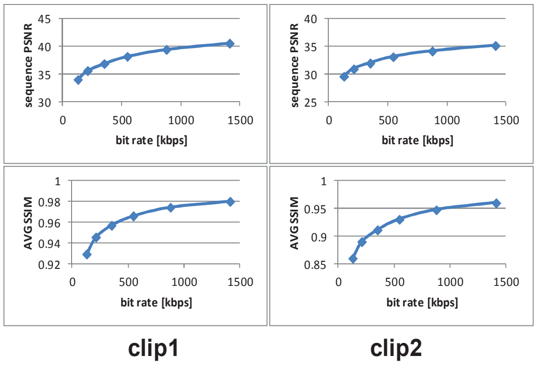

Figure 7: PSNR/SSIM values for the clips encoded in the six bit rates

E. Model comparison

The chunk-MOS model provides the best performance. It provided the best performance for the tablet and for the phone for one of the clips, and second best performance for the phone for the other clip in case the model parameters are optimally tuned per (clip, device). However of equal importance, and as very relevant for unseen content, it also showed to be less sensitive to non-optimal parameter settings. Indeed, the model is not very sensitive to using other parameter (β and γ) settings, to separating the profiles in a training and a test set, and to using parameters settings based on other (clip, device) pairs. Setting β=0.32 and γ=0 provides good results for the 4 (clip, device) pairs for which we have subjective data. One disadvantage is that the MOS values per quality level are required to apply the model, which may not always be the case.

The SSIM model performed best for one of the phone cases in optimally trained parameter settings, but showed to be extremely sensitive to parameter settings based on other (clip, device) pairs. Therefore, the SSIM model is not accurate for unseen content (and to a lesser extent for unseen devices). The PSNR model performs clearly worse than the former models (chunk-MOS and SSIM model) and is also very sensitive to parameter setting.

The QL model gives good results when parameters are optimally trained, and is also moderately sensitive to other parameter settings. The BR model has the lowest performance as using the bit rate in a linear model is not a good measure for MOS.

VI. Conclusions

The chunk-MOS model has proven to be a good MOS estimator resulting in an accuracy of about 0.3 for tablet and 0.4 for phone in the given data set. The QL model is also a good MOS estimator, and is equivalent with the chunk-MOS model when the quality levels have been encoded at equidistant MOS. The accuracy of the model is currently (with the available data set) mainly limited by the error (uncertainty) of the obtained subjective MOS values in the dataset which are of similar order. The model itself has not yet reached its limits of accuracy. It is hard to predict the lower bound of accuracy that could be reached with less noisy data, but we expect that considerable further improvement is possible.

The chunk-MOS model is well positioned to be used in a video QoE assessment module that, by inspecting HTTP requests for HAS content, predicts the quality delivered to the user. The model can be used for individual users, but can also be the basis to give a quality indicator per cell or per larger geography.

VII. Future Directions

Strictly speaking, the results we obtained are valid for the type of content and RDA we have data for. Ideally we should collect more data for a wider range of content types. The model was tuned for the type of profiles that were observed with a Apple's HLS client. Other RDAs may lead to other typical profiles. It should be checked that in the region in the (μ,σ,ϕ) space where new profiles lie, the model still provides an accurate MOS prediction.

Acknowledgment

The information provided is based on work carried out in an on-going NGMN Project SERQU; the work in the NGMN Project SERQU is not finalized yet. Any information provided is preliminary and subject to change in a final approved NGMN Document on SERQU.

References

[1] Global mobile network traffic – a summary of recent trends", Analysis Mason report, 29 June 2011.

[2] "Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats", ISO/IEC 23009-1:2012.

[3] "HTTP Live Streaming Overview", Apple developer doc, 2011. https://developer.apple.com/library/mac/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/StreamingMediaGuide.pdf

[4] A. Zambelli, "ISS Smooth Streaming Technical Overview", Microsoft technical document, March 2009 http://download.microsoft.com/download/4/2/4/4247C3AA-7105-4764-A8F9-321CB6C765EB/IIS_Smooth_Streaming_Technical_Overview.pdf

[5] "Beginning Flash Media Server 4.5 – Part 6: Using HTTP Dynamic Streaming", Adobe technical document, http://www.adobe.com/devnet/adobe-media-server/articles/beginning-fms45-pt06.html

[6] T. Hossfeld, S. Egger2, R. Schatz2, M. Fiedler3, K. Masuch2, C. Lorentzen, "Initial delay vs. Interruptions: between the devil and the deep blue sea", In Proc. QoMEX (Quality of the Multimedia Experience) 2012, Yarra Valley, Australia, 2012.

[7] O. Verscheure , P. Frossard , M. Hamdi "User-oriented QoS analysis in MPEG-2 video delivery", Real-Time Imaging 5, pp. 305-314 (1999).

[8] J. Padhye, V. Firoiu, D. Townsley and J. Kurose, "Modelling TCP throughput: A simple model and its empirical validation" in Proc. SIGCOMM Symp. Communications Architectures and Protocols, pp. 304-314, Aug. 1998.

[9] S. Akhshabi, A.C. Begen, C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP", In proc. of the second annual ACM conference on Multimedia systems, pp. 157-168, February 23-25, 2011, San Jose, California.

[10] M. H. Pinson and S. Wolf, 'A new standardized method for objectively measuring video quality', IEEE Trans. Broadcast., vol. 50, no. 3, pp. 312–322, Sep. 2004.

[11] A. K. Moorthy and A. C. Bovik, "Visual importance pooling for image quality assessment," IEEE J. Sel. Topics Signal Processing, vol. 3, no. 2, pp. 193–201, Apr. 2009.