Monitoring YouTube QoE: Is Your Mobile Network Delivering the Right Experience to your Customers?

Pedro Casas*, Raimund Schatz*, and Tobias Hoßfeld†
*Telecommunications Research Center Vienna - FTW, Email: surname@ftw.at
†University of Würzburg, Institute of Computer Science, Email: hossfeld@informatik.uni-wuerzburg.de

***Abstract*—YouTube, the monster application of today's Internet, is changing the way ISPs and network operators manage quality monitoring and provisioning on their IP networks. YouTube is currently the most consumed Internet application, accounting for more than 30% of the overall Internet's traffic worldwide. Coupling such an overwhelming traffic volume with the ever intensifying competition among ISPs is pushing operators to integrate Quality of Experience (QoE) paradigms into their traffic management systems. The need for automatic QoE assessment solutions becomes even more critical in mobile broadband networks, where over-provisioning solutions can not be foreseen and bad user experience translates into churning clients. This paper presents a complete study on the problem of YouTube Quality of Experience monitoring and assessment in mobile networks. The paper considers not only the QoE analysis, modeling and assessment based on real users' experience, but also the passive monitoring of the quality provided by the ISP to its end-customers in a large mobile broadband network.**

***Index Terms*—Quality of Experience; YouTube; Real-time Traffic Monitoring; MOS; 3G/HSPA Networks.**

I. Introduction

YouTube is one of the most popular applications in today's Internet. On its own, YouTube accounts for more than 30% of the overall Internet's traffic [1], with 72 hours of video uploaded every minute and over 4 billion videos viewed every day [4]. This outstanding and ever-growing success imposes serious challenges for network operators, who need to engineer their systems to correctly handle the huge volume of traffic and the vast number of users in efficient ways. The issue becomes even more challenging for mobile network operators, who need to offer high quality levels to reduce the risks of clients churning for quality dissatisfaction, particularly in current highly competitive mobile broadband markets. The popularity of YouTube is also growing in mobile networks: the volume of traffic carried by YouTube flows in mobile devices has tripled in 2011, and more than 20% of the global YouTube views come today from mobile devices [4].

The research community has addressed the challenges imposed by YouTube's popularity from multiple perspectives: by characterizing its traffic [2], by studying the video delivery infrastructure [3], [8], by exploring the correlations between network and users behavior [2], [7], and by assessing the Quality of Experience (QoE) in controlled lab studies [11], [12] or in the field [13]. In this paper we present a complete study on the problem of YouTube Quality of Experience monitoring and assessment in mobile networks. The main objective of the paper is to guide network operators in getting an answer to a basic yet difficult to answer question: is my mobile network delivering the right QoE to those end-users watching YouTube videos? The challenges associated to this question are various and go beyond the classical approach of QoE analysis for network dimensioning, i.e., beyond determining the minimum bandwidth requirements such that the targeted QoE requirements are met.

QoE monitoring in YouTube faces several scientific challenges: firstly, a set of traffic descriptors or measurements which are accessible for the network operator and that correlate with the video quality experienced by the end-user must be identified. The closer to the end-user these measurements are performed, the easier the QoE monitoring task becomes. However, network monitoring at the edge of the network (i.e., at the end-users' terminals or set-top boxes) is difficult to realize due to scalability, privacy, and management issues, specially in the case of large-scale monitoring. For this reason, measurements should be performed at the core of the network, making even more challenging the QoE estimation task. Authors in [2], [7] have proposed to monitor the ratio between the encoding bitrate of the YouTube video and the throughput achieved by underlying network flows as a QoE indication. While this approach makes sense and is simple to apply, it can be too rough to distinguish between a good or bad experience due to the high variance of both metrics. This is specially true when the video bitrate and the network throughput are similar.

The second important challenge is on YouTube QoE modeling and assessment: appropriate QoE models which can map the identified measurements into QoE levels must be conceived. Defining these models is not a trivial task, and requires both controlled lab studies as well as analysis in real service conditions to obtain reliable results. The final challenge is associated to the implementation and deployment of the complete monitoring system in the core of the network, which requires a system capable of handling the extraction of the necessary measurements and the application of the QoE models, both in real-time. This paper addresses the aforementioned challenges in a comprehensive manner, presenting the analysis and the solutions to achieve all the steps from the modeling to the real-time monitoring of the QoE in YouTube. The focus is set on broadband mobile networks scenarios, which are more sensitive to bad QoE levels.

The remainder of the paper is organized as follows: Section II presents an overview of some selected studies on YouTube monitoring and QoE. Section III tackles the problem of QoE modeling in YouTube, considering both lab studies as well as measurements performed in the field. In Section IV we discuss and evaluate an off-line monitoring technique to map YouTube traffic flows into QoE levels, exclusively relying on the analysis of network packet traces. Section V integrates this technique into a real-time monitoring system for YouTube QoE, presenting evaluation results on its application at the core of a broadband mobile network. Finally, Section VI concludes this work.

II. Related Work

The evaluation of YouTube from the end-user perspective has been recently addressed in multiple works [2], [7], [9]–[13]. Authors in [2] present a YouTube performance and user experience analysis in terms of video startup latency and ratio between download rate and video encoding rate. Similarly, [7] presents an analysis on YouTube and DailyMotion performance, using the aforementioned ratio as a user-experience metric. Both studies perform the analysis from a pure network and application perspective, without considering the end user's opinions in terms of QoE.

The standard approach to analyze the QoE of a network application such as YouTube is to conduct controlled lab experiments [19]–[21]. The key benefits of such an approach rely on the full control the experimenter has on the overall evaluation process. However, lab experiments miss out many important QoE influence factors such as usage context, content preferences by individual users, or device usability among others, introducing differences w.r.t. evaluations conducted in the field. For this reason, experiments where the participants use the application in the real running environment are additionally conducted to improve the quality of the analysis.

The studies conducted in [9]–[13] do consider subjective user experience analysis in HTTP video streaming, directly asking the participants about their impressions on the perceived performance. In [9], a YouTube-like player is used to conduct subjective tests through crowdsourcing[1]. Results show that the number of stallings in a given period and the stallings' duration are the most impacting parameters on YouTube QoE. The study is complemented in [11], where the impacts of video startup latency are additionally studied through lab experiments. Authors in [10], [12] follow a similar approach to evaluate the QoE in HTTP video streaming. The aforementioned works limit their study to very controlled usage scenarios, which impacts the quality and generalization of the obtained results. In a very recent study we have analyzed the QoE of YouTube in a real mobile broadband network scenario [13], complementing the results of previous lab and crowdsourcing studies.

As regards monitoring, QoE monitoring in YouTube has very recently attracted the attention of the research community. The authors of [15] presented a client-side software tool to monitor YouTube traffic at the application layer, estimating the amount of video content at the YouTube playing buffer to predict stalling events on the video playback. The approach is interesting but not applicable in the case of an ISP willing to monitor YouTube QoE on its network, since the installation of additional software on the client side is not a practical option. In [16], [17], authors present an approach for measuring YouTube stallings in an off-line fashion, relying on network packet traces; the study presents some first interesting results, but the validation of the technique is limited to a couple of application examples, making it difficult to draw conclusions on the accuracy of the approach. Finally, we have recently introduced a Deep Packet Inspection (DPI) technique to estimate the number of stalling events and their duration in a YouTube video [18], and tested its accuracy with off-line YouTube video packet traces. In this paper we build upon our previous studies and give a further step on the direction of QoE monitoring and assessment, presenting the first results in the large-scale, real-time monitoring of YouTube QoE in mobile broadband networks.

[1]Crowdsourcing in this context means outsourcing the subjective evaluation tasks to a vast and highly distributed group of people, who perform the evaluation in their own computers and are paid for it.

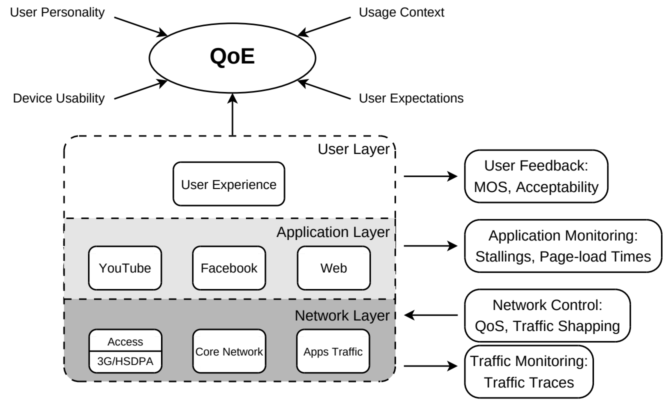


Fig. 1. Layered QoE modeling and assessment methodology for networking applications.

III. Modeling QoE in YouTube

The experience of a user with any application is conditioned by multiple influence parameters, including dimensions such as technical characteristics of the application, user personality and expectations, user demographics, device usability, and usage context among others. Particularly when evaluating networking-based applications such as YouTube, the influence of the network itself as well as its interplay with the particular application have to be linked to the user's opinions, additionally identifying those perceivable performance parameters that are most relevant to the user experience. This mapping is realized by analyzing and correlating the three layers depicted in figure 1: the *network layer* accounts for the influence of the network QoS parameters (e.g., network bandwidth, RTT, etc.); the *application layer* considers both the technical characteristics (e.g., video bit-rate) and the perceivable performance parameters of the application (e.g., page-load times, video stallings, etc.); finally, the *user layer* spans the user subjective opinions on the evaluated application (e.g., MOS values, acceptability, etc.).

In the case of YouTube QoE, previous studies [9], [12] have shown that both the number of stalling events and their duration are the only impairments visible to the end-user. A stalling event in HTTP video streaming applications corresponds to the interruption of the video playback due to the depletion of the playback buffer at the user's terminal. When the available bandwidth is lower than the required video bitrate, the playback buffer becomes gradually empty, ultimately leading to the stalling of the playback. For this reason, a good starting point to assess the QoE in YouTube is to study the relations between both the number and the duration of these stalling events and the users' perception. Having a model which can map stallings to QoE has a very powerful advantage, that of becoming independent of the underlying specific characteristics of the network in which the YouTube QoE will be monitored.

Figures 2 and 3 depict these relations for both controlled studies (lab and crowdsourcing) and field experiments we have recently performed in [9], [11], [13]. In the case of lab and crowdsourcing studies, 37 participants watched different YouTube videos for which a fully controlled stalling pattern was applied (i.e., number and duration of stalling events were perfectly defined), and then rated the perceived overall quality according to an ordinal ACR Mean Opinion Score (MOS) scale [19], ranging from "bad" (MOS=1) to "excellent" (MOS=5). The obtained results are depicted in figure 2. In the case of field studies, a group of 33 participants used mobile broadband 3.5G modems connected to the 3G HSPA network of a leading European network operator to watch their preferred YouTube videos on their own laptops, rating the overall perceived quality. Stalling patterns can not be controlled in field studies; for this reason, participants' traffic was rate-limited to different down-link bandwidth values, and the resulting stallings were measured at the application layer using the client-side software tool developed in [15]. The reader should note that this tool was installed in the laptop of the participants for this specific study, but that using such a client-side tool in a large-scale and distributed mobile network scenario is not scalable in the practice. The obtained results are depicted in figure 3.

Both lab and field studies show that user perception of stalling events is highly non-linear, with one single stalling event already significantly impairing the overall experience. In both cases, a single stalling event reduces the video quality from excellent to fair (i.e., 1 MOS point in the scale). Note that the maximum ratings provided by users in both figure 2 and figure 3 are never 5 but somewhere between 4.3 and 4.6. This is a well known phenomenon in QoS studies, where users hardly employ the limit values of the scale for their ratings. A second stalling event has also a strong influence on YouTube QoE, but saturation already starts after 2 stallings, as even getting more than 4 stallings slightly reduces the QoE from around 2 to 1.6. Stallings duration also plays an important role in YouTube QoE, but shows to be less critical in this case. For example, doubling the stalling duration from 2 to 4 seconds in the lab studies has a limited impact, but increasing

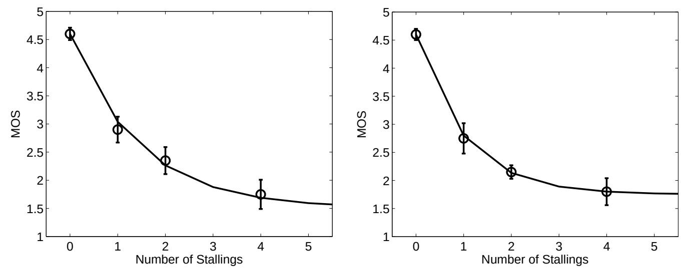


Fig. 2. MOS vs number of stallings from Lab and Crowdsourcing measurements: stallings of 2 (left) and 4 (right) seconds of duration.

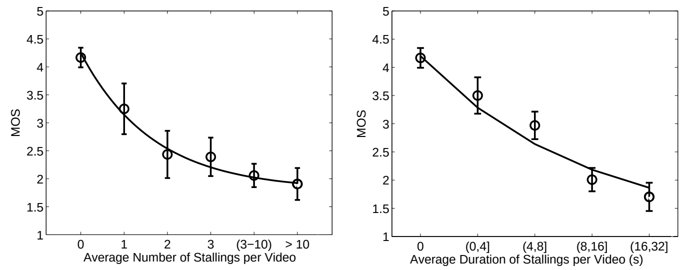


Fig. 3. MOS vs average number (left) and average duration (right) of stallings per video from field measurements.

its value to more than 8 seconds shows deterioration of the user experience in the field.

In section V we combine the results obtained from both studies into a single model that maps the stalling patterns that occur in the playback of a YouTube video into the QoE as perceived by the user watching the video. The question that now arises is how can we actually measure these stalling patterns without having access to the YouTube application running at the terminal of the user. We tackle this issue in the next section.

IV. From Packets to YouTube QoE

As previously stated, YouTube QoE is determined by the stalling patterns in the video playback as experienced by the end user. In [18] we have recently introduced a Deep Packet Inspection (DPI) technique that permits to reconstruct these stalling patterns from the packets transmitted in the YouTube IP flows. The basic idea of this technique is to estimate the playing time that is accumulated in the buffer of the YouTube player, comparing the playback times of the video frames and the time stamps of the received packets. If the playback buffer runs empty, the video stalls until more packets are received.

The playback times of the video frames composing the video can be obtained by dissecting the metadata present in the so called *video container* (FLV, MP4, etc.). Each YouTube video is compressed and encoded as an FLV, MP4, etc. file which is a container format for media files. The container includes the compressed video and audio, as well as the information needed by the YouTube player to decode and display the video content. The header of these media files starts with a well-defined signature identifying the corresponding container format, and contains metadata information such as the times when the video frames have to be actually displayed. The YouTube player opens a new TCP connection each time it downloads a new FLV, MP4, etc. file or if the user jumps to another time in the video. The developed DPI technique consists in identifying the beginning of a new YouTube video flow as marked by the signature of its container, and extracting the corresponding play times of the downloaded content to estimate the accumulated video play time at the buffer.

Let us define some additional parameters that compose this DPI technique. The first and most important parameter is the total downloaded video play time τ_i, which is updated from every new TCP ack received at time t_i. As we said before, the value of τ_i is obtained from the parsing of the video container metadata. We additionally define the play time ρ_i and the stalling time σ_i, which are the user experienced video play time and stalling time after the i-th TCP ack. The amount of buffered video time is indicated as β_i, and it corresponds to the difference between the downloaded video play time τ_i and the actually played time ρ_i, i.e., $\beta_i = \tau_i - \rho_i$. We also consider a boolean stalling variable ψ_i, which indicates whether the video is currently playing ($\psi_i = 0$) or stalling ($\psi_i = 1$).

In addition, the YouTube player uses two different playing and stalling thresholds to control the way it consumes video frames from the playback buffer. The first threshold Θ_0 defines the minimum amount of buffered video time that has to be exceeded to start playing a stalled video; the second threshold Θ_1 specifies the minimum amount of buffered video time necessary to continue playing a video once the playback has started. So if we consider the video buffer size β_{i-1} at time t_{i-1}, then we get that if β_{i-1} exceeds Θ_0, the video starts playing; on the other hand, if the video buffer falls bellow Θ_1, then the video stalls. Hence, stalling occurs if the following condition is true: $(\psi_{i-1} \wedge (\beta_{i-1} < \Theta_0)) \vee (\neg\psi_{i-1} \wedge (\beta_{i-1} < \Theta_1))$. The measurement studies performed in [18] revealed that these two buffer thresholds can be reasonably taken as $\Theta_0 = 2.2$ s and $\Theta_1 = 0.4$ s. While these two thresholds are not constant and depend on the specific characteristics of a video, results show that even if using not 100% exact values the estimations can be very accurate. Using these definitions, the stalling pattern of a YouTube video over time can be obtained as follows:

$$\psi_i = \psi_{i-1} \wedge (\beta_{i-1} < \Theta_0) \vee \neg\psi_{i-1} \wedge (\beta_{i-1} < \Theta_1) \quad (1)$$

$$\sigma_i = \sigma_{i-1} + \begin{cases} t_i - t_{i-1}, & \text{if } \psi_i \\ 0, & \text{if } \neg\psi_i \end{cases} \quad (2)$$

$$\rho_i = \rho_{i-1} + \begin{cases} 0, & \text{if } \psi_i \\ t_i - t_{i-1}, & \text{if } \neg\psi_i \end{cases} \quad (3)$$

$$\beta_i = \tau_i - \rho_i \quad (4)$$

Finally, as depicted in eq. (2) and eq. (3), the time elapsed between the previous ack at time t_{i-1} and current ack at time t_i increases the play time ρ_i or the stalling time σ_i, depending on the resulting video state (i.e., playing or stalling). Since YouTube first starts buffering (i.e., stalling state) until the threshold Θ_0 is exceeded, the iterative computation of the different variables is initialized with $\sigma_0 = \rho_0 = 0$ and $\psi_0 = 1$.

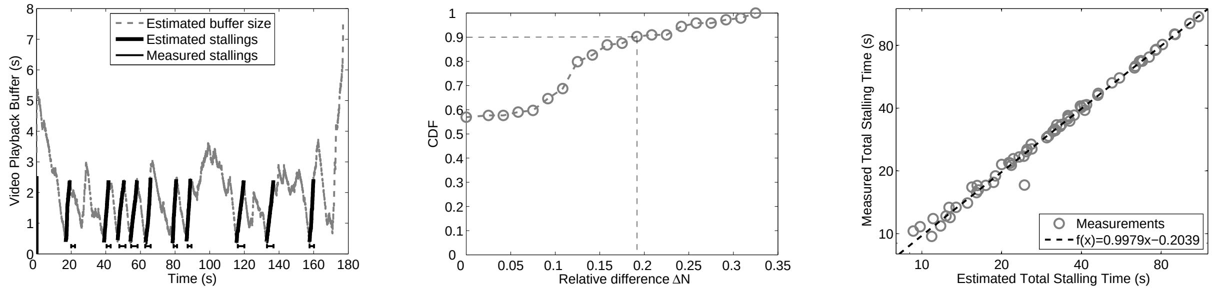
Figure 4(a) shows an exemplary case of the estimated video buffer size and stalling events over time of a YouTube video, using the described technique. The video starts playing as soon as Θ_0 is exceeded. However, when the buffer is below Θ_1, the video stalls. In addition, the stalling pattern as measured on the application layer is plotted as gap lines, which shows that the estimated and the actually observed stallings fit quite well. However, there are some small differences caused by different aspects. Firstly, we rely on TCP acknowledgments, which might be delayed due to network performance fluctuations and/or protocol implementation issues (e.g., if using delayed acks). Secondly, the video buffer thresholds are average values over a large set of videos, but the actual thresholds for an individual video depends on its particular characteristics (e.g., the sequence of I, B, P frames). Hence, small differences to the considered values may emerge for some videos.

Figures 4(b) and 4(c) present the estimated stalling patterns obtained for a set of 100 YouTube videos downloaded under different bandwidth conditions. The study is performed in an off-line fashion, by analyzing the packet traces captured during the video downloading/playback. Figure 4(b) shows the relative difference $\Delta N = \frac{|N_e - N_a|}{N_a}$ between the number of stallings N_a measured on the application layer and the estimated N_e through the DPI technique. The relative difference is small and below 20% for 90% of the videos. The reader should note that the minimum relative difference ΔN obtained in the case of estimation errors for $N_a = 1, 2,$ or 3 is of $1/3 = 33\%$, which actually shows that the errors depicted in 4(b) occur for bigger values of N_a. As we showed in section III, a difference of 1 or more stalling events after 3 or 4 stallings has a negligible impact on YouTube QoE, reducing the impacts of such estimation errors. As regards stallings' duration, a comparison of the estimated stalling time for these videos as depicted on figure 4(c) reveals an almost perfect match with the actual stalling time as measured on the application layer.

These results show that the DPI technique can actually be used to extract the stalling patterns that occur during the streaming of a YouTube video, which can then be mapped to QoE values by applying the models depicted in section III. The main limitation of this estimation technique as presented so far is that it has not been conceived as a tool for monitoring the QoE of YouTube from the perspective of an operator, who actually needs to run such estimations in the core or close to it to have an idea of the overall quality his customers are experiencing. The last step to achieve such a monitoring system is described in the next section.

V. YOUTUBE QOE MONITORING: HOW GOOD IS YOUR MOBILE NETWORK DOING?

The main question that this paper tries to answer is how to actually determine how good is a certain mobile network in providing YouTube to its customers with good QoE levels. In this section we build upon the QoE models depicted in section III and on the aforementioned DPI technique to build and show example results on the application of a real-time monitoring



(a) Estimated buffer size from packet traces. (b) Estimated number of stalling events. (c) Estimated stalling time.

Fig. 4. Estimated stalling patterns from packet traces. Results presented in (b) and (c) correspond to the estimation performed for 100 YouTube videos.

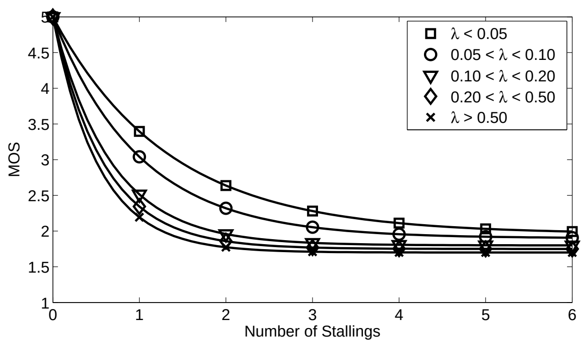


Fig. 5. MOS vs number of stallings, depending on the fraction λ of total stalling duration.

system capable of assessing the YouTube QoE in the core of a mobile broadband network.

We have devised an optimized DPI-based technique for reconstructing the stalling patterns of YouTube videos from network packets captured and analyzed on the fly. The basic functioning of this technique relies on the steps described in previous section. Going into the particular implementation details of the approach is out of the scope of this paper; nevertheless, we provide some basic details to guide the reader and to facilitate the interpretation of the obtained results. The technique is implemented as a module of a 3G mobile network monitoring system, which basically captures all the data packets flowing from/to the terminals of the mobile users to/from the Internet and performs analysis on the fly, in a stream basis. Traffic is continuously monitored at the Gn interface of the mobile network, and in the particular case of YouTube QoE monitoring, the system issues a report or *ticket* every $T = 60$ seconds for every YouTube flow detected in the stream of packets. Among other information, each ticket contains the estimated YouTube QoE MOS value for the 60 seconds and the number of video seconds that were consumed with that MOS. The estimated YouTube QoE is computed from the extracted number and duration of stallings in the 60 seconds time slot. Using such time slots of short duration permits to have a clear idea of the performance of the network as regards YouTube QoE in a real-basis, providing valuable information for the network operator on the satisfaction of his customers.

In order to map the extracted number and duration of stalling events into MOS values, we have adapted the datasets and curves presented in section III to the specific slotted time functioning of the monitoring system. In particular, we have considered a new mapping function where we take the ratio λ between the total stalling time and the total video elapsed time (i.e., playing + stalling time) in the corresponding time slot as a better image of the impacts of stalling time on YouTube QoE. This permits to limit the effects of videos with different durations, as we are now considering the stalling time relative to the length of the evaluation (i.e., the length of the time slot). The resulting YouTube stallings–QoE mapping model depicted in figure 5 is decomposed in five different functions, depending on the value of λ computed in the time slot of length T. The five functions have all the same shape, in the form of:

$$\mathrm{MOS}(n)_i = a_i \cdot e^{-b_i \cdot n} + c_i, \quad \forall i = 1, 2, 3, 4, 5. \tag{5}$$

where n is the number of stalling events estimated on the time slot of length T and $\{a_i, b_i, c_i\}$ depend on the computed value for λ. At every new time slot where a YouTube video is detected (active or starting), the value of λ is obtained as follows: first, compute the total stalling time σ and the total play time ρ for this time slot; then, if the total video elapsed time $\rho + \sigma$ is smaller than the length of the time slot T, then compute $\lambda = \sigma/(\sigma + \rho)$; otherwise, $\lambda = \sigma/T$. The curves depicted in figure 5 deserve some clarifications: firstly, the MOS value computed for $n = 0$ stallings only makes sense for the curve in which $\lambda < 5\%$; in all the other cases, $n > 0$. Secondly, the curves only show mappings for up to $n = 6$ stallings; this is because a YouTube video with more than such a number of stalling events can be directly declared as very bad quality, and no extra mapping is therefore required.

In order to validate the resulting on-line YouTube QoE monitoring system and the corresponding mappings, we replay some of the network packet traces captured in the field trial study conducted in [13], for which we have the MOS values declared by the users as ground truth. Figure 6 (left) compares both the declared MOS and the predicted MOS values for 16 different videos which experienced different stalling patterns in the field trial. All the considered videos have a total

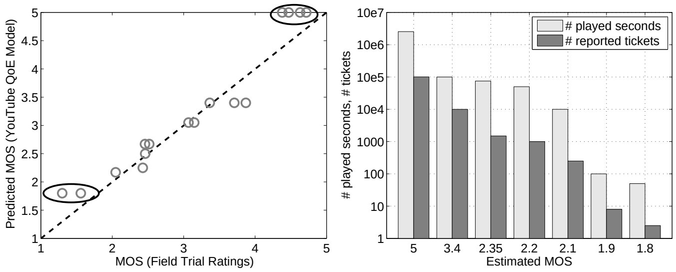


Fig. 6. YouTube QoE real-time monitoring results. (left) Validation with real traces from the field trial. (right) The monitoring is performed at the Gn interface of the 3G HSPA network of a leading European network operator, on a period of 1 hour.

duration of less than 60 seconds, just to avoid any biased comparison due to the different evaluation procedure used in the field trial and on this evaluation. Obtained results are very accurate and close to the actual declared MOS values by the participants, but some strange deviations occur at the edges of the rating scale, both at very low or very high MOS values. This difference comes from the edge-ratings phenomenon previously mentioned in section III. In the field study, ratings for 0 stallings correspond to MOS values around 4.5, while the model depicted in figure 5 gives a MOS value of 5 on these situations. Similarly, the limit values for very bad quality provided by the model are slightly higher than the actual opinion of the users; for this reason, the model provides a MOS value around 1.8 when users actually rate around 1.5. In any case, the reader should note that none of both identified differences are an issue to consider.

To conclude with this work, we present in figure 6 (right) the YouTube QoE monitoring results obtained by using the described real-time monitoring system with the real mobile broadband traffic of a leading European network operator. The monitoring is performed on 1 hour of on-line traffic flows observed at the Gn interface. The histogram depicts both the number of reported tickets and the total played seconds at the different YouTube QoE levels provided by the devised model. To avoid misunderstandings, the reader should note the logarithmic scale on the y-axis of the plot. The most important comment on these results comes from the fact that with this system, it is actually possible to have a clear view of the performance of the mobile network as regards the satisfaction of the customers consuming YouTube videos. As regards the specific MOS values, the estimated YouTube QoE is excellent for more than 90% of the tickets and of the video time consumed during the analyzed hour. For about 8% of the issued tickets and 5% of the total video time, the quality achieved was average (i.e. MOS = 3.4 in this case). Regarding bad quality estimation, one of the main limitations of doing only monitoring is that the system can not say whether bad quality events come from problems on the network or in any other place (the customer terminal, the YouTube servers, a bad SNR, etc). In this particular evaluation, the aggregated occurrence of bad quality events is practically negligible.

VI. Concluding Remarks

In this paper we have studied the problem of YouTube QoE real-time monitoring and assessment in mobile networks. We have addressed all the different steps to reach a system capable of giving concrete real-time indications on the performance of a mobile broadband network regarding the experience of the customers watching YouTube videos. In particular, we have covered the modeling of YouTube QoE by combining results from both lab and field studies; we have studied the problem of how to extract YouTube performance indicators related to the QoE perceived by end-users, relying exclusively on packet-level measurements; and most important, we have devised and depicted evaluation results that show the potential and feasibility of doing real-time QoE monitoring in services such as YouTube in mobile broadband networks. This paper provides a first answer to the original question we posed at the very beginning: Is Your Mobile Network Delivering the Right Experience to your Customers? Now we can answer it.

Acknowledgments

This work has been performed in the framework of the projects ACE 2.0 and U-0 at the Telecommunications Research Center Vienna (FTW), and has been funded by the Austrian Government and the City of Vienna through the program COMET. The authors would like to thank Eduard Hasenleithner and Tobias Witek for their highly valuable participation in all the implementation details of the study.

References

[1] V. Gehen, A. Finamore, M. Mellia, and M. Munafo, “Uncovering the Big Players of the Web”, in *TMA’12*, 2012.

[2] A. Finamore et al., “YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience”, in *ACM IMC’11*, 2011.

[3] R. Torres, A. Finamore, J. Kim, M. Mellia, M. Munafo, and S. Rao, “Dissecting Video Server Selection Strategies in the YouTube CDN”, in *IEEE ICDCS’11*, 2011.

[4] YouTube Press Room - Statistics, http://www.youtube.com/t/press_statistics, accessed on October 2012.

[5] Y. Liu, K. Gummadi, B. Krishnamurthy, and A. Mislove, “Analyzing Facebook Privacy Settings: User Expectations vs. Reality”, in *ACM IMC’11*, 2011.

[6] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. Bhattacharjee, “Measurement and Analysis of Online Social Networks”, in *ACM IMC’07*, 2007.

[7] L. Plissonneau and E. Biersack, “A Longitudinal View of HTTP Video Streaming Performance”, in *ACM MMSys’12*, 2012.

[8] S. Alcock et al., “Application Flow Control in YouTube Video Streams”, in *ACM SIGCOMM Computer Communication Review*, vol. 41 (2), pp. 24-30, 2012.

[9] T. Hossfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, and P. Tran-Gia, “Quantification of YouTube QoE via Crowdsourcing ”, in *IEEE MQoE 2011*, 2011.

[10] R. Mok, E. Chan, X. Luo, and R. Chang, “Inferring the QoE of HTTP Video Streaming from User-Viewing Activities”, in *ACM SIGCOMM W-MUST’11*, 2011.

[11] T. Hossfeld et al., “Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea”, in *QoMEX 2012*, 2012.

[12] R. Mok, E. Chan, and R. Chang, “Measuring the Quality of Experience of HTTP Video Streaming”, in *IFIP/IEEE IM 2011*, 2011.

[13] P. Casas, A. Sackl, S. Egger, and R. Schatz, “YouTube & Facebook Quality of Experience in Mobile Broadband Networks”, in *IEEE QoEMC 2012*, 2012.

[14] S. Hemminger, “Network Emulation with NetEm”, in *LCA 2005*, 2005.

[15] B. Staehle, M. Hirth, R. Pries, F. Wamser, and D. Staehle, “YoMo: A YouTube Application Comfort Monitoring Tool”, in *EuroITV 2010*, 2010.

[16] S. Rugel, T. Knoll, M. Eckert, and T. Bauschert, “A Network based Method for Measurement of Internet Video Streaming Quality”, in *Proc. European Teletraffic Seminar*, 2011.

[17] M. Eckert and T. Knoll, “An Advanced Network based Method for Video QoE Estimation based on Throughput Measurement”, in *Proc. EuroView 2012*, 2012.

[18] R. Schatz, T. Hoßfeld, and P. Casas, “Passive YouTube QoE Monitoring for ISPs”, in *Proc. FingNET 2012*, 2012.

[19] International Telecommunication Union, “Methods for Subjective Determination of Transmission Quality”, *ITU-T Recommendation P.800*, 1996.

[20] International Telecommunication Union, “Subjective Video Quality Assessment Methods for Multimedia Applications”, *ITU-T Recommendation P.910*, 2008.

[21] International Telecommunication Union, “Estimating End-to-End Performance in IP Networks for Data Applications”, *ITU-T Recommendation G.1030*, 2005.