App for Dynamic Crowdsourced QoE Studies of HTTP Adaptive Streaming on Mobile Devices

Michael Seufert, Nikolas Wehner, Pedro Casas
Austrian Institute of Technology GmbH, Vienna, Austria
{michael.seufert.fl | nikolas.wehner.fl | pedro.casas}@ait.ac.at

***Abstract*—This demonstration introduces CroQoE, a new app for crowdsourced QoE studies of HTTP Adaptive Streaming. The app is running on the mobile device of the study participant and connects to a backend server to control the subjective QoE study. As video content is a crucial QoE factor, the proposed app allows the participant to submit keywords for topics or contents, which videos he wants to watch during the study. Matching videos are retrieved and prepared on the backend server, and the test conditions are included. Thereby, the user is able to watch interesting and relevant content during the QoE study.**

I. Introduction and Related Work

Quality of Experience (QoE) is the subjective perception of the quality of a networked service as a whole by the end user. It is influenced by different QoE factors, which are characteristics of the user, system, service, application, or context [1]. In order to understand what factors influence the QoE of a service, extensive subjective studies have to be conducted. Several QoE studies have already been conducted for HTTP Adaptive Streaming (HAS) [2], which is the currently prevailing video streaming technology, and one of the most popular and most demanding Internet services. However, these studies rather focused on technical parameters of the streaming as a first step, and found that initial delay, stalling, and visual quality have a large impact on the QoE. Still, the high dimensional space of QoE factors including all human, system, and context factors was not exhaustively investigated yet to obtain a holistic QoE model for HAS.

Recently, crowdsourcing, i.e., the outsourcing of small tasks to a large crowd, has been widely used for subjective QoE studies. The advantages of crowdsourcing over laboratory studies are its price, speed, and the more realistic setting of service consumption, especially, in terms of context and system factors. One problem of previous crowdsourced QoE studies is that they were almost exclusively designed for the desktop environment. Participants of the study accessed the web-based test frameworks with their PC or laptop in order to watch different video clips and rate the test conditions [3]. This confines the applicability of the resulting QoE models to that context. In particular, it does not allow to easily transfer or generalize to the context of mobile video streaming. Although mobile devices account for the majority of video requests from the popular streaming service YouTube[1], only few mobile QoE apps exist. Moreover, these apps do not allow to conduct a controlled QoE study with predefined test conditions, but only monitor the streaming behavior. YoMoApp [4] is additionally able to collect subjective feedback, but most other mobile QoE apps only try to estimate QoE based on monitored QoE factors without asking subjective ratings of users, e.g., [5].

Another typical problem of QoE studies is that test content is selected by researchers, and not by the users. The reason is that the displayed video clips have to be prepared to include the test conditions (such as stalling events or quality adaptation). However, the content selection might introduce a bias on the ratings, as users are used to select the content themselves. Thus, users are well aware that they are not consuming the video service naturally, but rather in the context of a QoE study. It might even happen that participants have to watch content that they do not like, which might cumulate not only in biased ratings but in an abandonment of the QoE study.

This work presents a new app, named CroQoE, for crowdsourced QoE studies of HAS. Its benefit are twofold. First, it allows to conduct QoE studies on mobile devices in order to investigate also the context of mobile video streaming. Thereby, the app includes all features of traditional desktop-based crowdsourcing QoE frameworks [6], e.g., downloading of video content before the study, reliability checks of user interactions, and provision of verification codes for payments on crowdsourcing platforms. Second, users can submit topics for video clips, in which they are interested. If video clips of that topic are available, the test conditions are included and the dynamically prepared clips are used in the QoE study. Thereby, the app can also be used for adaptive crowdsourcing [7], i.e., test conditions can be selected based on dynamic criteria, e.g., the certainty of QoE scores. The CroQoE app and the demonstration are described in Section II, and Section III outlines future works.

II. CroQoE

The front-end of CroQoE runs on the users' mobile device as an application and is responsible for the user interaction and the presentation of the QoE study. The back-end server is responsible for the preparation of the video content and storage of the data of the QoE study.

a) Front-end app: The user interface of the app has different states depending on the user's progress, which the app negotiates with the back-end at the app start. First, the user has to submit the crowdsourcing credentials (e.g., worker ID, study ID), as well as topics that correspond to his interests. If no topics were submitted, random or default videos of the study would be used for the further process. After uploading

[1] https://www.youtube.com/yt/about/press/ - accessed May 8, 2018.

the request to the back-end, the app is immediately closed and changes into waiting state.

As soon as the back-end prepared the test conditions, the user is notified with a push notification (shown in the notification bar on the user's device) that the videos are ready for download. Now, CroQoE shows a download button and progress bar of the download, which turn into a button to start the QoE study. All videos are played out in full-screen mode and landscape orientation. During the playback, media controls are deactivated and proper test execution is monitored. After each video, a dialog is shown and the user has to submit ratings on subjective quality, quality acceptance, and content liking.

When all test conditions were rated, a verification code is computed from the crowdsourcing credentials and the collected data are uploaded to the back-end server. The uploaded data are composed of the crowdsourcing credentials, the video IDs, the test conditions applied to the video clips, the monitored data about the test execution, the submitted ratings, and the verification code. When the upload is done, the app displays the redeemable verification code to the user.

b) Back-end server: The back-end server of CroQoE is responsible for dynamically preparing the test conditions. It receives the user's keywords and uses the API of a big video database to crawl matching videos according to some guidelines, e.g., only popular videos in HD with a duration of less than five minutes. In order to avoid that users have to watch the same video several times, a random video ID is chosen out of the returned video IDs. The corresponding video files are subsequently downloaded or copied. Note that each study could specify mandatory video content or use default or random clips when the users do not enter a topic.

Next, the selected video files are manipulated with FFmpeg to include the test conditions. The test conditions can be fixed or can be selected dynamically from a given pool of test conditions. First, the videos are cut to the desired length and split according to the tested QoE pattern. Then, the different QoE influence factors of HAS like initial delay, stalling, or changes of the visual quality are adjusted. Initial delay or stalling are generated by adding additional still image sequences of the desired length to the original video. These sequences simply show one frame with a GIF overlay portraying the buffering symbol. Different visual qualities are imitated by re-encoding the whole video clip or parts of it with a new video bit rate or by changing the quantization parameter. Finally, all split parts of the clips are concatenated.

When the test sequences are prepared, the user is notified via push notification. Therefore, CroQoE utilizes Google's Firebase Cloud Messaging (FCM)[2], which works across multiple different platforms, e.g., Android and iOS. For the identification of the devices, the FCM SDK sets up a Firebase token that registers the client app instance. All user identification and communication between app and back-end in CroQoE is done with this token. The content on the server is linked to the user's Firebase token so that the push notification is sent explicitly to the user device with the corresponding token and the prepared content is only accessible by this device. Finally, the uploaded data of the user is stored in a database for later evaluation of the QoE study.

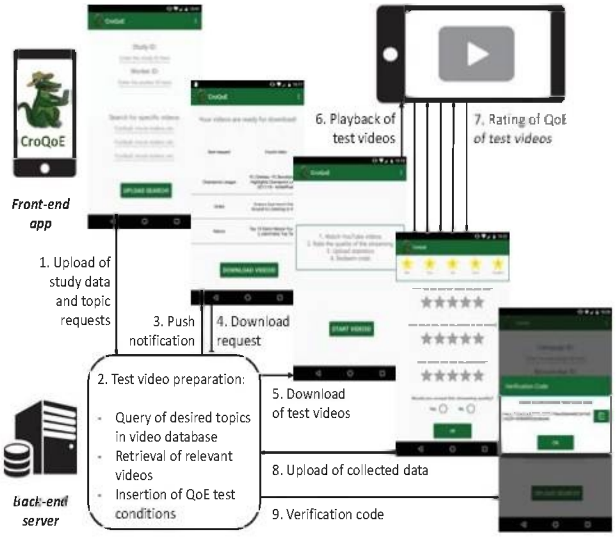
CroQoE
Front-end app
1. Upload of study data and topic requests
3. Push notification
4. Download request
6. Playback of test videos
7. Rating of QoE of test videos
2. Test video preparation:
- Query of desired topics in video database
- Retrieval of relevant videos
- Insertion of QoE test conditions
5. Download of test videos
8. Upload of collected data
9. Verification code
back-end server

Fig. 1: Procedure of CroQoE demonstration

c) Demonstration: In the demonstration, the procedure of CroQoE during a typical QoE study of HAS is shown. An Android version of CroQoE can be used to specify desired topics, as well as to watch and rate the prepared video clips. The preparation of the clips, the inclusion of test conditions at the local Linux back-end server, and all communication between front-end and back-end is shown on a dashboard. The procedure is illustrated in Figure 1.

III. Outlook to Future Work

While the more natural streaming consumption on mobile devices is clearly an improvement for the design of crowdsourced QoE studies of HAS, the proposed methodology of CroQoE to let users select the content of the QoE study has to be thoroughly investigated. It has to be evaluated if the ratings of test conditions can be merged when they are based on different contents, and when different users see different content. The desired outcome is that there is less noise and smaller variance in the resulting QoE ratings, because the liking of the content is improved by the content selection (higher liking with smaller variance) and does not/less negatively affect the QoE ratings. However, an explicit QoE study with CroQoE still has to be conducted in future work.

References

[1] P. Le Callet *et al.*, "Qualinet White Paper on Definitions of Quality of Experience," Tech. Rep., 2013.

[2] M. Seufert *et al.*, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communications Surveys & Tutorials*, 2015.

[3] T. Hoßfeld *et al.*, "Survey of Web-based Crowdsourcing Frameworks for Subjective Quality Assessment," in *MMSP*, 2014.

[4] M. Seufert *et al.*, "Unsupervised QoE Field Study for Mobile YouTube Video Streaming with YoMoApp," in *QoMEX*, 2017.

[5] Q. A. Chen *et al.*, "QoE Doctor: Diagnosing Mobile App QoE with Automated UI Control and Cross-layer Analysis," in *IMC*, 2014.

[6] T. Hoßfeld *et al.*, "Best Practices and Recommendations for Crowdsourced QoE - Lessons learned from the Qualinet Task Force Crowdsourcing," Tech. Rep., 2014.

[7] M. Seufert *et al.*, "Impact of Test Condition Selection in Adaptive Crowdsourcing Studies on Subjective Quality," in *QoMEX*, 2016.

[2] https://firebase.google.com/docs/cloud-messaging/ - accessed May 8, 2018.