UNIVERSITÄT AUGSBURG
Universitätsbibliothek

OPEN ACCESS
OPUS AUGSBURG

On the analysis of youTube QoE in cellular networks through in-smartphone measurements

Sarah Wassermann, Pedro Casas, Michael Seufert, Florian Wamser

Angaben zur Veröffentlichung / Publication details:

Wassermann, Sarah, Pedro Casas, Michael Seufert, and Florian Wamser. 2019. "On the analysis of youTube QoE in cellular networks through in-smartphone measurements." In *12th IFIP Wireless and Mobile Networking Conference (WMNC), 1-13 September 2019, Paris, France*, edited by Merouane Debbah, Guy Pujolle, Zoubir Mammeri, Marilia Curado, Thi Mai Trang Nguyen, and Selma Boumerdassi, 71–78. Piscataway, NJ: IEEE. https://doi.org/10.23919/wmnc.2019.8881828.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
Deutsches Urheberrecht
Weitere Informationen finden Sie unter: / For more information see:
https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/

On the Analysis of YouTube QoE in Cellular Networks through in-Smartphone Measurements

Sarah Wassermann*, Pedro Casas†, Michael Seufert‡ and Florian Wamser‡
* Inria Paris, † AIT Austrian Institute of Technology, ‡ University of Würzburg
sarah.wassermann@inria.fr, pedro.casas@ait.ac.at, {michael.seufert, florian.wamser}@informatik.uni-wuerzburg.de

***Abstract*—Cellular-network operators are becoming increasingly interested in knowing the Quality of Experience (QoE) of their customers. QoE measurements represent today a main source of information to monitor, analyze, and subsequently manage operational networks. In this paper, we focus on the analysis of YouTube QoE in cellular networks, using QoE and distributed network measurements collected in real users' smartphones. Relying on YoMoApp, a well-known tool for collecting YouTube smartphone measurements and QoE feedback in a crowdsourcing fashion, we have built a dataset covering about 360 different cellular users around the globe, throughout the past five years. Using this dataset, we study the characteristics of different QoE-relevant features for YouTube in smartphones. Measurements reveal a constant improvement of YouTube QoE in cellular networks over time, as well as an enhancement of the YouTube video streaming functioning in smartphones. Using the gathered measurements, we additionally investigate two case studies for YouTube QoE monitoring and analysis in cellular networks: the machine-learning-based prediction of QoE-relevant metrics from network-level measurements, and the modeling and assessment of YouTube QoE and user engagement in cellular networks and smartphone devices. Last but not least, we introduce the YoMoApp cloud dashboard to openly share smartphone YouTube QoE measurements, which allows anyone using the YoMoApp smartphone app to get immediate access to all the raw measurements collected at her devices.**

***Index Terms*—Mobile Network Measurements; Quality of Experience; YouTube; Crowdsourcing; Machine Learning.**

I. Introduction

Today, access to the Internet is primarily carried out through cellular networks, and smartphones are the most common way to consume Internet content, from web browsing and video streaming to a plethora of novel services offered through apps. The increase in the volume and heterogeneity of content accessed in cellular networks and smartphones forces cellular-network Internet Service Providers (ISPs) to improve their performance monitoring and assessment capabilities, in particular with respect to understand the performance as perceived by their customers. The Quality-of-Experience (QoE) paradigm permits to understand and assess the functioning of networks and services from the eyes of the end user. QoE-based network measurements represent today a main source of information for general network operation and management.

In this paper, we address the problem of YouTube-QoE monitoring and analysis in cellular networks following a data-driven approach, by analyzing a dataset of crowdsourced measurements on YouTube QoE, passively collected in real users' smartphones. According to the official statistics of YouTube, more than half of YouTube views today come from smartphone devices, thus the relevance of our study.

The dataset is built from measurements collected with the YoMoApp tool [1], an app that we have conceived in the past to monitor network and QoE-relevant metrics related to YouTube directly at the smartphone. YoMoApp is publicly available and can be directly installed through the Google Play Store. Using YoMoApp, we collected measurements related to more than 3000 YouTube sessions worldwide, streamed on 70 different cellular-network providers to more than 360 different customers, between 2014 and 2018. The YoMoApp tool passively gathers multiple QoE-relevant metrics and network-performance indicators related to YouTube, including measurements at the player side (e.g., stalling events, changes in video resolution, initial delay), the network side (throughput, downlink/uplink bytes, radio access technology, etc.), as well as at the user side, retrieving user feedback through QoE surveys displayed by the app after completion of a session.

The analysis of this dataset reveals interesting findings on the QoE of YouTube in cellular networks along the studied time period, including: (i) **a sustained QoE improvement of YouTube streaming in smartphones**, (ii) **an enhanced performance of the YouTube video streaming protocol**, and (iii) **a positive impact of these improvements on the engagement of the users watching YouTube videos**, with an increase of over 30% on the video watching time.

Our work is not only unique in terms of the richness of the measurements we analyzed, but also in terms of the different perspectives from which we look at the overall problem. In particular, we additionally investigate two problems linked to the monitoring and analysis of YouTube mobile QoE: first, we build machine-learning-based prediction models to estimate application-layer QoE-related metrics collected by YoMoApp from network-layer measurements only, which allows to extend and generalize the YouTube-mobile monitoring approach without requiring users to actually use YoMoApp. Predictions also include user engagement as a paramount target. Second, we study the performance of multiple QoE-assessment models previously proposed in the literature and standards, contrasting their outcomes to the real user feedbacks.

Our last contribution is on opening the monitoring platform offered by YoMoApp to the network measurement community. In particular, we introduce the YoMoApp cloud dashboard for openly sharing the full, raw measurements retrieved by YoMoApp on registered devices. In a nutshell, through this

dashboard, one can register a YoMoApp instance installed at an Android device and get instant access to the measurements collected on it. This has many implications and benefits for those interested in the problems tackled in this paper; for example, one could install multiple devices with YoMoApp and run any sort of measurement campaigns to perform tasks such as network-performance assessment, cellular-ISP benchmarking, network monitoring, and many more.

Current work builds on top of our recent paper on YouTube QoE analysis in smartphones [6], extending the data-characterization part as well as the analysis through the application of machine-learning models and QoE-based assessment techniques. As such, this paper offers a comprehensive perspective on the problem of YouTube-QoE monitoring and analysis in cellular networks through the eyes of the end user, and presents highly relevant use cases for machine-learning-based data analytics in networks. The work is complete and unique in terms of the addressed perspectives of the problem, from the data collection, characterization, and analysis, to the application of QoE modeling and machine-learning techniques to enable a broader visibility on YouTube QoE in cellular networks and smartphone devices.

The remainder of the paper is organized as follows: Section II reports on related papers, in particular around the topic of YouTube QoE in cellular networks and smartphones. Section III describes in detail the YoMoApp application and the YoMoApp cloud dashboard for open data collection and sharing, and explains the main concepts behind the measurements. Section IV presents and discusses the results of the data characterization and analysis, particularly studying the behavior and improvement of QoE-relevant metrics during the five years spanned by the dataset. Section V studies in detail two interesting applications of the YoMoApp system for network monitoring and analysis, including the prediction of QoE-relevant metrics and the modeling and assessment of user QoE and user engagement. Finally, Section VI presents some future outlook and concludes this work.

II. Related Work

QoE-relevant KPI monitoring has been widely addressed in the literature, mostly focused on fixed networks and considering in-network or network-side measurements. In [2], authors provide an overview on QoE-based network-monitoring approaches and their associated challenges. Focusing on the problem of QoE monitoring for video streaming, there are different proposed techniques and models translating in-network measurements and/or in-device application measurements to QoE-relevant metrics. Among multiple mapping models studied in the literature, we refer to a recently standardized QoE assessment model for adaptive video streaming – ITU-T P.1203 [4], which predicts the Mean Opinion Score (MOS) of a video session from direct analysis of both network- and application-layer measurements.

In [5], we have proposed YoMo, an in-device, application- and DPI-based tool for YouTube-QoE monitoring, capturing video player activity and buffering conditions to infer re-buffering events. In [7], [8], [9], we extended YoMo and its overall concept to monitor YouTube QoE in cellular and fixed-line networks at scale, using DPI approaches. Others [10], [11], [12] adopted similar in-application measurements for YouTube-QoE monitoring, relying on application-side tools to directly collect KPIs such as playback delay, re-buffering events, video resolution, or quality switches. Application-side monitoring provides accurate measurements for QoE assessment, as these can be directly observed, without the need of additional estimation or mapping approaches.

The wide adoption of end-to-end encryption has turned previous DPI-based approaches unreliable or even unfeasible, motivating a surge of papers focusing on the analysis of in-network measurements through machine-learning models. For example, in [3], [13], authors apply different machine-learning approaches to estimate QoE-relevant metrics for YouTube by extracting features from the stream of encrypted packets, using simple features such as packet times and sizes, or throughput. Similarly, authors in [14] follow a machine-learning-based analysis to infer QoE metrics for YouTube streaming over cellular networks. Other recent papers propose to reconstruct the evolution of the buffered video playtime [7], but analyzing the encrypted stream of packets through heuristics and statistical modeling approaches [15].

When dealing exclusively with cellular networks and smartphones, there are many tools to monitor QoE-relevant KPIs, including Netalyzr [17] and Mobilyzer [16]. Smartphone-app QoE can be monitored with QoE Doctor [18], an active-measurement tool analyzing both network and application features. Other tools for measuring YouTube QoE in smartphones are introduced in [20] and [19]. YoMoApp [1] is an extension of our previous YoMo tool, but implemented as an Android app to passively measure YouTube QoE-relevant features in smartphones. Last, previous papers have also presented results on machine learning for QoE prediction in smartphones: our previous work [22], [23] as well as [21] use machine-learning models to infer the QoE of smartphone apps, relying on in-device and/or in-network measurements.

III. Using YoMoApp for YouTube Mobile Analysis

YoMoApp [1] provides a distributed monitoring platform for YouTube QoE, collecting user feedback in a crowdsourced manner, and passively measuring a large set of QoE-relevant KPIs at the player and network side. Metrics such as stallings, initial playback delay, and video resolution are retrieved at play time. All YoMoApp measurements are periodically uploaded to a remote server, building a comprehensive database of YouTube performance- and QoE-related measurements. It is possible to access these measurements for further analysis using the YoMoApp dashboard service, which is presented and described below. Next, we describe the basic concepts of YoMoApp, the considered KPIs, and the incentives offered to users to motivate them using the app.

Log-file type	Parameters				
Data	Current playtime	Buffer	Available playtime		
Events	Video ID	Quality	Network	Received bytes	Transmitted bytes
	Cell ID	Signal	SSID	BSSID	RSSI
	Location	Title	Duration	Screen orientation	Player size
	Player mode	Volume	MSE	Supported codecs	Player state
	Dialog	Content rating	Quality rating	Streaming rating	Acceptability rating
	YouTube loading time	Advertisement	Video end	App behavior	Hyperlink
Statistics	Date	Time	Device ID	Mobile operator	Country
	Network switches	Networks	Screen size	Screen density	Orientation changes
	Orientations	Player resizes	Player sizes	Handovers	Cell ID
	Video ID	Video title	Log time	Length of video	User engagement
	Initial delay	Quality switches	Qualities	Stalling events	Total stalling time
	Average stalling time	Maximum stalling time	Average buffer	Maximum buffer	Pause events
	Content rating	Quality rating	Streaming rating	Acceptability rating	

Table I: Monitored KPIs per log file in YoMoApp.

A. YoMoApp Basics and KPIs

YoMoApp is an Android app (freely available at the Google Play Store) which replicates the original YouTube app in functionality and design. An Android WebView is embedded to display the YouTube mobile website, using an HTML5 `video` element relying on adaptive-streaming technology for the video playback. Additional functions perform the monitoring of the application-level parameters in the application. The monitoring is done at runtime via JavaScript, which queries the HTML5 `video` element.

We use JavaScript event listeners to monitor changes of the player state (e.g., playing, paused, buffering, ended), and the resolution of the `video` element. The app monitors the current playback time and the buffered playtime every second. Additionally, we retrieve metadata, e.g., the YouTube video ID, title, duration of the watched video. The gathered data is then sent to and processed by the Android app. As the usage of JavaScript is prone to inconsistencies and errors, e.g., missing/incorrect values or non-equidistant data queries, the data is post-processed locally by YoMoApp.

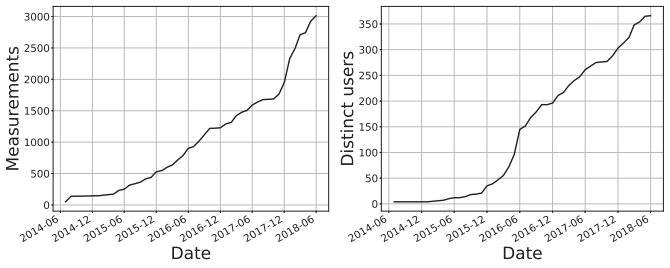
Besides playback events, YoMoApp measures both network and context features. Moreover, it collects device features such as size of the screen, orientation, playback audio volume, size of the player, and playing mode (e.g., full screen). Lastly, the application gathers network-traffic statistics such as per-second uploaded/downloaded bytes, as well as information such as GPS-based location, cellular operator, ID of the cell, Radio Access Technology (RAT), or strength of the signal.

YoMoApp additionally collects QoE feedback provided by the user, once a video is fully watched or aborted. A simple questionnaire with multiple questions allows the user to rate the QoE of the video session according to a standard ACR MOS scale [24], ranging from bad (MOS = 1) to excellent (MOS = 5). Questions include the user's feedback on the quality of the video, the quality of the streaming, the user's opinion on the video content, as well as the service acceptability (yes/no). The QoE-feedback questionnaire is presented to the user only if she wishes to provide such feedback, which is specified at the YoMoApp starting time. All these measurements and user-QoE feedbacks are structured over three different text files, logged for each video session: the **data log file** keeps those metrics related to the playback buffer of the video; the **events log file** tracks a rich number of KPIs which are reported in an event-based manner, including states of the video player, loading times, transmitted bytes, as well as all QoE-related answers from the user; last, the **statistics log file** is computed from the processing of both the data log and event log files, once a video session has ended. This log aggregates multiple application- and network-performance KPIs, such as stallings, initial playback delay, quality switches, and many more. The file also includes other metadata related to the video session, such as mobile operator and handovers, cell IDs, or video content details. Log files are identified by a unique session ID, which includes a device ID, the session date, and the starting time.

A complete list of the KPIs collected for the individual log files are presented in Table I. Measurements belonging to data and event log files are synchronized through Unix timestamps. KPI monitoring is done either at every second – for example, when tracking the video playtime – or just when a specific event happens, such as a change in the played video resolution. In contrast, the statistics log file offers an overview/aggregation of the video streaming session. Further details on the measurements collected through YoMoApp can be found at the official YoMoApp documentation – http://yomoapp.de/documentation.pdf.

B. Incentives and the YoMoApp Dashboard

The large-scale usage of in-device monitoring tools such as YoMoApp is subject to the incentives a user receives to install such tools on his smartphone. Measurement tools operating at the end devices are more useful to the ISP when these are installed and used at large scale, offering representative and meaningful information. To provide incentives for using YoMoApp, the user can access several aggregate statistics for each of her video sessions. A QoE map-view is also included within the app, which displays all the geolocalized QoE ratings and the corresponding network operators, for the full set of collected ratings across all YoMoApp users. Results can be displayed by ISP, allowing the user to compare, through easy-



(a) Cumulative # of sessions. (b) Cumulative # of users.

Figure 1: Number of sessions and distinct users over time.

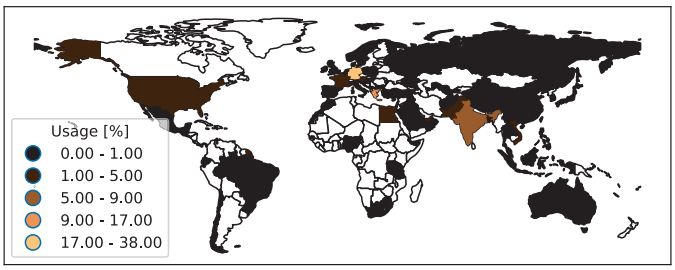


Figure 2: Worldwide usage of YoMoApp.

to-understand heatmaps, which operator performs best in certain locations in terms of YouTube mobile QoE performance – see http://www.yomoapp.de for examples.

A second and strong incentive for using YoMoApp is introduced in this paper: the YoMoApp cloud dashboard, available at http://yomoapp.de/dashboard. Through this dashboard, users can access at any time the aforementioned log files containing all the raw measurements and KPIs collected by YoMoApp at their own devices. A user has access to the data retrieved at any device for which she has the YoMoApp device ID (available through YoMoApp), by simply creating a user account at the dashboard, and associating all the YoMoApp device IDs she has access to. There is no limit on the number of different devices a user can associate to her user account, turning YoMoApp and the dashboard into a powerful distributed monitoring platform for YouTube mobile measurements analysis. We stress again the fact that the data which can be accessed through the dashboard includes the full, raw, fine-grained measurements collected by YoMoApp as described in Table I. This is highly useful for deep analysis on multiple relevant problems associated to YouTube mobile video streaming in the wild.

Besides full raw measurements access, the dashboard allows any user to browse over the complete database of measurements covering all YoMoApp users, in the form of aggregated and anonymized statistics, maps, and heatmaps, providing additional visibility. We are currently working on different incentive-driven approaches to allow and motivate users to install more YoMoApp instances and perform further measurements, as well as additionally sharing their own (anonymized) measurements with others; for example, we are testing an approach inspired on Peer-to-Peer (P2P)-based file sharing, providing access to anonymized measurements from other devices in equal volume to the measurements generated by the device(s) under the control of a user: the more measurements she generates with his devices, the more measurements she can access from devices of other users.

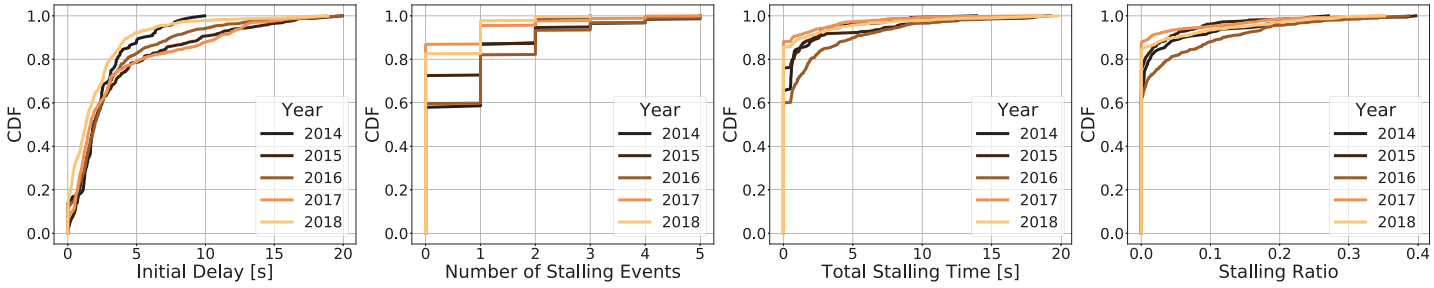
The combined usage of YoMoApp and the dashboard offers multiple network QoE monitoring and analysis opportunities to the network-measurement community: for example, it allows for field testing, distributed cellular-network performance monitoring, YouTube mobile QoE modeling and assessment in operational environments, analysis of the impact of different mobile-network technologies on YouTube mobile QoE, long term characterization of YouTube streaming strategies and even controlled QoE analysis. Next, we analyze the set of measurements so far collected through YoMoApp, and address some of the aforementioned monitoring and analysis directions.

IV. YouTube Mobile QoE Analysis

We now study the measurements collected with YoMoApp during the last five years. We analyze the evolution of YouTube in smartphones along time, regarding QoE metrics, user engagement and network performance. The dataset contains more than 3000 complete video sessions, captured between July 2014 and June 2018. Sessions correspond to 366 different users worldwide. Figs. 1a and 1b report the accumulated number of YoMoApp video sessions streamed over time and the cumulative number of unique devices, respectively. A surge of new users is clearly observed starting in 2016, which comes as a consequence of a stronger advertisement of YoMoApp and an increased dissemination through different research communities and conferences. The number of streamed sessions, new users, and collected measurements has more than doubled since January 2017. Interestingly, there were more than 900 new video sessions during the first half of 2018, largely exceeding the number of video sessions monitored in 2017. We conclude that the usage trend for YoMoApp is very positive.

The distribution of collected measurements worldwide is depicted in Fig. 2, in the form of a heatmap diagram. Measurements are distributed on 58 different countries. About 38% of the measurements come from Germany, 17% from Greece, 9% from India, and 5% from France. Measurements gathered in other countries represent a share equal or less than 3% each. We want to stress again that YoMoApp measurements are performed at the end devices of the users, using their corresponding mobile operators, resulting in a diverse set of measurements in terms of devices and network properties.

YouTube QoE in smartphones has improved over the past years: we now focus on the analysis of different QoE-related metrics along time. In particular, we study the evolution of: initial delays, re-bufferings, stalling times, and re-buffering ratios. Fig. 3 depicts the empirical distribution of these metrics per year. A first observation is that there is a clear enhancement of all QoE-related metrics along time, 2018 being the year with best performance in terms of initial playback delays and re-buffering events. Next, we also show that such an improvement is reflected by the QoE feedbacks reported by the end users (cf. Fig. 7). About 90% of the sessions in 2018



(a) initial delay (b) number of stallings (c) total stalling time (d) stalling ratio

Figure 3: Temporal evolution of the performance of YouTube mobile streaming in terms of QoE-relevant KPIs.

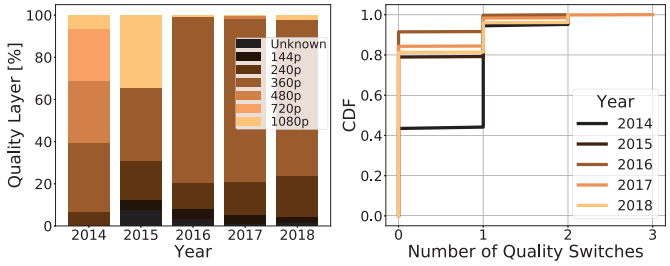
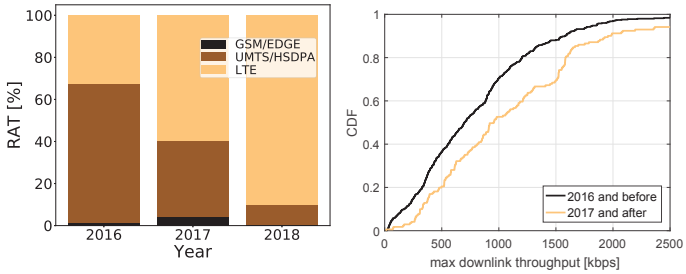


Figure 4: Video-quality levels and quality switches.



(a) Radio Access Technology. (b) Max. downlink throughput.

Figure 5: Radio access and video download throughput.

have an associated initial delay below 5 seconds, and a similar fraction corresponds to videos streamed and displayed without stalling. On the other hand, initial delay for video sessions in 2016 was below 5 seconds for about 80% of the videos, and only 60% of the videos were displayed without re-buffering events. Furthermore, more than 15% of the videos in 2016 suffered from a re-buffering ratio higher than 10%, whereas this fraction falls to about 5% in 2017/2018.

YouTube mobile video distribution is more efficient today than in the past: the played out video quality levels grouped by year and the distribution of the number of quality switches per year are illustrated in Fig. 4. The distribution of requested video qualities by the YoMoApp video player reveals that, in contrast to the period from 2016 to 2018, the played out video qualities varied much more back in 2014 and 2015, with a higher prevalence of higher quality levels as compared to today. The YouTube streaming service has been evolving over time, not only for the fixed-line network scenario, but mainly in mobile networks. When YouTube started playing in mobile devices, the adaptive-streaming policy was less conservative and higher quality levels would be requested in adaptive streaming mode. From 2016 onwards, the most dominant video quality changed to 360p, which is a more conservative quality level, imposing less bandwidth requirements. There are also videos with lower video qualities like 144p or 240p, but almost no HD content was streamed within the last three years with YoMoApp. This is perfectly aligned to our previous findings on YouTube QoE in smartphones [25], where we observed that lower vertical resolutions result in the same subjective experience as higher resolutions when dealing with smartphones, due to the small screen sizes. Thus, it makes less sense and is less efficient to stream HD content on YouTube in smartphones.

As a consequence, it is also not surprising that the number of quality switches observed within the last three years is much lower compared to 2014 and 2015. Fig. 4 displays the distribution of the number of quality switches per session. In more than 80% of the sessions, no quality switch could be observed for the period of 2016 to 2018, meaning that the initial quality selected by YouTube was matching the underlying network performance. In contrast, in 2014, only 43% of the sessions showed no quality switch, around 53% observed one quality switch, and the remaining sessions resulted in two or more quality switches.

Mobile network technology and performance have also improved, potentially resulting in increased user engagement: the distribution of the underlying RAT per year is displayed in Fig. 5a. We differentiate between 2G (GSM/EDGE), 3G (UMTS/HSDPA) and 4G (LTE). RAT information started being collected only from 2016 on. In 2016, UMTS/HSDPA was the dominant RAT, with a prevalence of about 66% of all sessions with cellular access. In 2017, the balance shifted and LTE became the dominant RAT with a share of 59%. This dominance increased even more in 2018, where sessions with LTE make up to 90% of all streaming sessions with cellular access. As a consequence, we observe better network performance over time. For example, Fig. 5b shows the distribution of the maximum download throughput achieved by YoMoApp video sessions before and after December 2016. The average max. download throughput increased from about 2Mbps to more than 10Mbps, and the median has also increased from

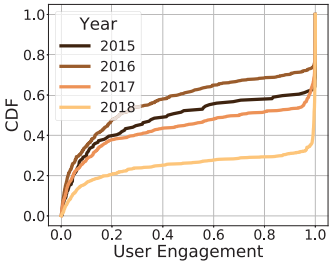
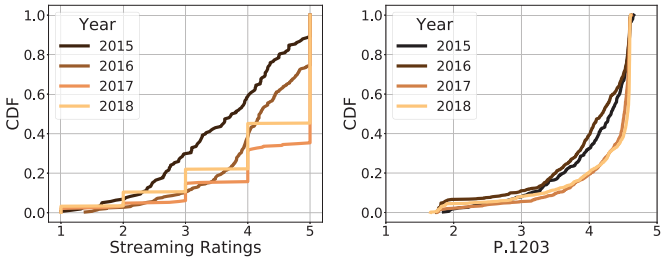
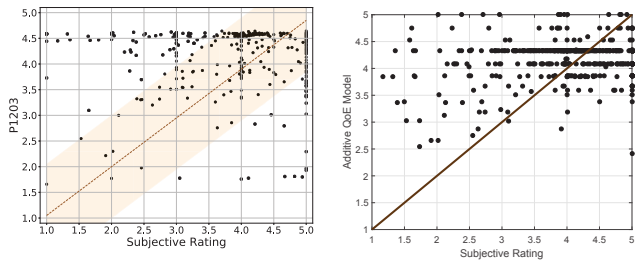


Figure 6: Evolution of user engagement.



(a) Subjective ratings. (b) P.1203 scores.

Figure 7: Distribution of MOS scores per session.



(a) P.1203 vs. subjective ratings. (b) QoE model vs. user feedback.

Figure 8: Modeled MOS scores vs. actual user feedback.

about 600kbps to 1Mbps.

User engagement is defined as the fraction of the total video length a user watched before the video was aborted or the video ended (100% user engagement). The user-engagement distribution per year is depicted in Fig. 6. User engagement started being measured in 2015, we therefore have no results for 2014. Results show how user engagement has systematically increased over time, and significantly in 2018. More than 60% of the videos were watched completely and only 20% of the users aborted the video at 20% or less of the video playback. This indicates that YoMoApp is increasingly being used as a standard video player. The increased user engagement can also be explained by the improvement of the network performance in terms of higher downlink throughputs. We note that video-duration averages and distribution are similar across the different years, ruling out potential bias on user engagement.

V. YouTube Monitoring with YoMoApp

We now show how YoMoApp can be used for network monitoring and YouTube QoE analysis purposes, tackling two different and highly relevant problems. Firstly, we focus on the problem of YouTube QoE modeling and assessment, calibrating different YouTube QoE models available in the literature as well as standardized models – in particular the ITU-T P.1203 model for adaptive video streaming [4]. We compare the outputs obtained from these models to the actual QoE feedbacks provide by YoMoApp users. Secondly, we tackle the prediction of QoE-relevant metrics as well as user experience and user engagement through the application of machine-learning models, using only network-layer measurements as input. As we said before, such machine-learning models enable an extended monitoring of YouTube mobile QoE, as one could still obtain the KPIs currently collected by YoMoApp, but without even requiring to use the app – for example, by just having a general purpose network-measurement app running in the background of the device.

A. QoE Modeling and Assessment

Fig. 7 depicts the distribution of (a) the subjective MOS scores as provided by the users and (b) an estimation of the MOS scores, obtained by the P.1203 model. Recall that QoE feedback is provided in terms of MOS scores, using a 5-level ACR scale [24]. For the sake of comparison to other objective QoE models, we focus on the QoE of the users regarding their opinion on the video streaming performance. As observed in Fig. 7a, MOS scores were reported by the users through a continuous scale before 2017, and using a discrete scale from 2017 on. As reported before, there is a clear QoE improvement during the last two years, with more than 80% of the videos rated with MOS scores equal or above 4; this fraction drops to a value between 40% to 60% in previous years. As shown in Fig. 7b, results are accurately captured by the P.1203 model predictions.

Besides the application of the P.1203 model, we additionally investigate simpler QoE models available in the literature [26], [27]. These models are of exponential nature, under the form $f(\alpha,\beta,\gamma,\delta,L,N) = \alpha \times e^{-(\beta \times L+\gamma)\times N}+\delta$, where α, β, γ, and δ are parameters that need to be calibrated through the specific dataset under analysis, and L and N correspond to the average stalling length and number of stallings respectively. We take a simple manual calibration approach to set $\alpha = 4$ and $\delta = 1$ to be within MOS range, and set the other parameters by a non-linear-least-squares regression.

Another family of models we look at are referred to as simple additive QoE models, which are expressed as a linear combination of individual models: $Q(x_1,\ldots,x_n) = \sum_{i=1}^{n} w_i \times Q_i(x_i)$, where weights w_i are ≥ 0 and $\sum_{i=1}^{n} w_i = 1$ [26]. We rely on non-linear-least-squares regression to determine the values of the parameters to tune. Our evaluation revealed that the additive QoE model expressed as $0.49 \times (4 \times e^{-0.14 \times \#stallings} + 1) + 0.17 \times (4 \times e^{-47.7 \times initialDelay} + 1) + 0.34 \times (4 \times e^{-0.44 \times \#qualitySwitches} + 1)$ is the one which fits best the data.

Fig. 8 depicts two scatter plots reporting the modeled MOS scores vs. the actual user subjective feedbacks. Fig. 8a reports the results for the P.1203 model, whereas Fig. 8b considers the best model of the two tested ones [26], [27], which corresponds to the additive QoE model. In both cases, we

observe that both QoE models tend to overestimate the actual QoE ratings reported by the users. This suggests that users might be actually more annoyed than what one could perceive by directly using these QoE models in practice.

Last, Fig. 9 depicts the linear correlations observed between both the subjective ratings and the P.1203 estimations and application-layer metrics such as stalling, initial delay, quality switches and up to user engagement. While correlations tend to be rather low, there is a clear negative impact of stalling duration, initial delay, and number of stallings on both QoE values (feedback and P.1203), as observed in past studies.

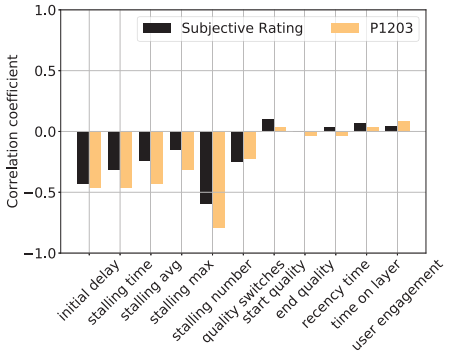


Figure 9: Linear correlations – subjective ratings and P.1203.

B. QoE Prediction through Machine Learning

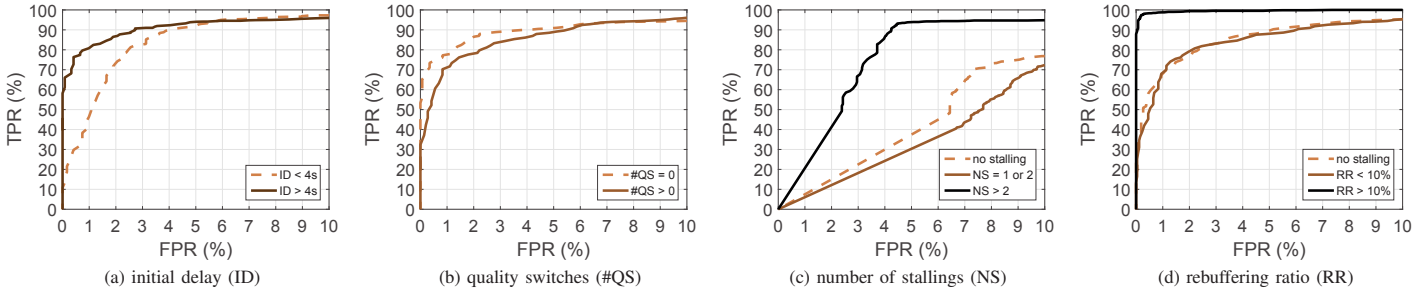
We focus now on the prediction of QoE-relevant metrics which are normally measured directly by YoMoApp, but assuming only access to the smartphone general network-level measurements, available through the Android APIs. The rationale is that we would like to monitor YouTube mobile KPIs such as initial delay, stalling, quality switches, QoE (MOS scores), and even user engagement, but without using an app like YoMoApp. These predictors could be applied in a more generic smartphone-based monitoring system, where users would not be forced to run an app with an embedded player such as YoMoApp to measure relevant KPIs, and where such KPIs could be actually forecasted for any user watching YouTube videos at her smartphone, independently of the YouTube player being used.

We tackle the prediction of four QoE-relevant metrics, the prediction of the MOS scores (as provided by the P.1203 model), and the prediction of the user engagement. We build predictors using machine-learning models, treating each problem as a classification task, where targets are discretized. The targets are as follows: (i) whether initial delays (ID) are above or below a pre-defined QoE-relevant threshold – based on previous work on initial delay tolerance, we set this value to 4 seconds; (ii) whether a video quality switch has occurred during the session or not (cf. Fig. 4); (iii) the number of stalling events (NS), considering three classes – *zero-stalling*, *mild-stalling*: one or two stalling events, and *severe-stalling*: more than two stallings; and (iv) the stalling frequency or re-buffering rate (RR), considering again three classes – *stalling-free*; *mild-stalling*: stallings occurred and lasted for less than 10% of the total duration of the video session, and *severe-stalling*: stallings occurred for more than 10% of the whole video session. For the prediction of QoE scores, we use as target a binary discretization of the MOS scores provided by the P.1203 model, and consider a two-class classification problem, either better or worse than MOS = 4. Finally, we turn the prediction of user engagement into a three-class classification task, predicting whether a user has watched less than 50% of the video, between 50% and 70%, or more than 70%. For each metric, we evaluate a random forest model with 10 trees through 10-fold cross validation. We rely on simple bootstrapping techniques to balance classes for learning purposes.

For all these prediction tasks, we rely on the network-layer features captured by YoMoApp, which can actually be measured by simply accessing the Android developer APIs. The full feature set encompasses 275 features, including information about the received signal strength, the number of handovers, the number of network switches, and multiple statistics about the incoming and outgoing traffic, aggregated at different time windows of 1, 5, 10, 30, and 60 seconds. The traffic is measured on three different levels: the total traffic transmitted/received by the device, the traffic captured over the mobile network, and the traffic sent/received by the application itself. We use feature-selection techniques to identify the most relevant features for each target. We find that about 30 features out of the 275 are needed to obtain highly similar accuracies to the ones achieved with the full feature set.

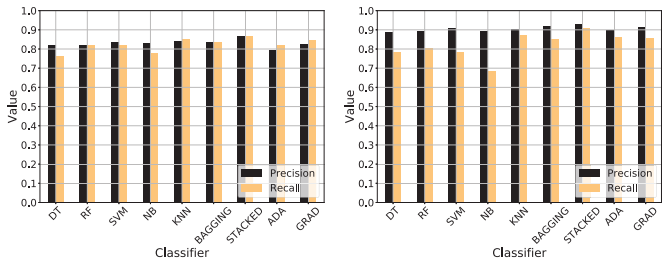
Fig. 10 reports the obtained results for the prediction of the four QoE-relevant KPIs in terms of ROC curves. ROC curves help understand the performance of binary-classification models at all classification thresholds and show the different false positive rates (FPRs) and true positive rates (TPRs). Our results are fairly accurate for the four prediction targets, achieving good classification rates for most of the classes. For example, the initial delay discrimination as well as the quality-switching detection can be done with a false positive rate below 5% for more than 90% of the sessions. Results are even better when predicting the re-buffering ratio, with an almost perfect performance for detecting bad-quality sessions with a high stalling ratio. Inferring the exact number of stalling events is clearly more challenging.

For the prediction of user engagement and MOS scores, we also consider random forests, but additionally evaluate other models such as a single decision tree (DT), SVM, *k*-nearest neighbors (KNN), and Naïve Bayes (NB). We also consider ensemble learning approaches, covering the three basic paradigms available in the ensemble-learning domain: bagging, boosting (AdaBoost (ADA) and gradient boosting (GRAD)), and stacking. Instead of constructing the most accurate model to interpret the data, ensemble learning approaches combine multiple models to improve analysis performance. Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Models built this



(a) initial delay (ID) (b) quality switches (#QS) (c) number of stallings (NS) (d) rebuffering ratio (RR)

Figure 10: QoE-metrics prediction performance. ROC curves evidence high recall for the considered classes.



(a) Prediction of user engagement. (b) Prediction of P.1203 MOS.

Figure 11: Prediction of user engagement and P.1203 MOS.

way are in general more robust to uncertainties and noise in the data, which helps in generalizing the obtained results.

Fig. 11 summarizes the obtained results in terms of precision and recall for all the tested models, obtained through 10-fold cross-validation. As before, high prediction performance can be achieved for both targets, particularly when using more complex, ensemble-based approaches, like stacked trees (STACKED). Prediction of P.1203 MOS classes and user-engagement discrimination can be realized with an overall accuracy of around 90%.

VI. Conclusion

In this paper, we have studied the problem of YouTube mobile QoE monitoring and analysis in a data-driven manner, by relying on a very rich and fairly large dataset of QoE measurements passively collected at users' smartphones with the YoMoApp monitoring framework. We introduced and discussed the different YoMoApp tools which grant open access to highly rich measurements retrieved at mobile devices. Through the analysis of these measurements, we are able to observe a systematic performance and QoE improvement of YouTube in mobile devices since 2014 till today, additionally evidencing that these enhancements might have a direct impact on the user engagement in YouTube mobile. We have additionally studied and discussed different network monitoring and analysis problems which can be tackled by relying on YoMoApp, showing its great potential. In particular, we presented different machine-learning approaches to monitor and predict QoE-relevant metrics for YouTube in smartphones, as well as to predict user engagement and QoE, using as input only those measurements which can be directly accessed through Android APIs – i.e., without the need of accessing any application-level KPI to perform the analysis. Besides noting the good performance of random forest models for QoE prediction, we have also presented evidence on the advantages of relying on more complex, ensemble techniques, to properly predict end user QoE and engagement.

References

[1] F. Wamser et al., "YoMoApp: A tool for analyzing QoE of YouTube HTTP adaptive streaming in mobile networks," in *EuCNC*, 2015.

[2] W. Robitza et al., "Challenges of future multimedia QoE monitoring for internet service providers," *Mmedia Tools and Apps*, vol. 76(21), 2017.

[3] I. Orsolic et al., "A machine learning approach to classifying YouTube QoE based on encrypted network traffic," *Mmedia Tools and Apps*, vol. 76(21), 2017.

[4] ITU, "ITU-T Recommendation P.1203: Parametric Bitstream-based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services over Reliable Transport," 2016.

[5] B. Staehle et al., "YoMo: A YouTube Application Comfort Monitoring Tool," in *QoEMCS*, 2010.

[6] N. Wehner et al., "Beauty is in the Eye of the Smartphone Holder: A Data Driven Analysis of YouTube Mobile QoE," in *CNSM*, 2018.

[7] R. Schatz et al., "Passive YouTube QoE Monitoring for ISPs," in *FINGNet*, 2012.

[8] P. Casas et al., "Monitoring YouTube QoE: Is Your Mobile Network Delivering the Right Experience to Your Customers?" in *IEEE WCNC*, 2013.

[9] P. Casas et al., "YOUQMON: A System for On-line Monitoring of YouTube QoE in Operational 3G Networks," *ACM SIGMETRICS PER*, vol. 41(2), 2013.

[10] H. Nam et al., "YouSlow: A Performance Analysis Tool for Adaptive Bitrate Video Streaming," *ACM SIGCOMM Computer Communication Review*, vol. 44(4), 2014.

[11] H. Nam et al., "QoE Matters More Than QoS: Why People Stop Watching Cat Videos," in *IEEE INFOCOM*, 2016.

[12] H. Nam et al., "YouSlow: What Influences User Abandonment Behavior for Internet Video?," Columbia University, Tech. Rep., 2016.

[13] M. H. Mazhar et al., "Real-time Video Quality of Experience Monitoring for HTTPS and QUIC," in *IEEE INFOCOM*, 2018.

[14] G. Dimopoulos et al., "Measuring Video QoE from Encrypted Traffic," in *IMC'16*.

[15] T. Mangla et al., "eMIMIC: Estimating HTTP-based Video QoE Metrics from Encrypted Network Traffic," in *TMA*, 2018.

[16] A. Nikravesh et al., "Mobilyzer: An open platform for controllable mobile network measurements," in *MobiSys*, 2015.

[17] C. Kreibich et al., "Netalyzr: Illuminating the edge network," in *IMC*, 2010.

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

[19] I. Ketykó et al., "QoE Measurement of Mobile YouTube Video Streaming," in *MoViD*, 2010.

[20] G. Gómez et al., "YouTube QoE Evaluation Tool for Android Wireless Terminals," *EURASIP Jour. on Wireless Comm. and Net.*, vol. 2014(164), 2014.

[21] V. Aggarwal et al., "Prometheus: Toward Quality-of-Experience Estimation for Mobile Apps from Passive Network Measurements," in *HotMobile*, 2014.

[22] P. Casas et al., "Predicting QoE in cellular networks using machine learning and in-smartphone measurements," in *QoMEX*, 2017.

[23] P. Casas et al., "Next to You: Monitoring Quality of Experience in Cellular Networks From the End-Devices," *IEEE TNSM*, vol. 13(2), 2016.

[24] ITU, "ITU-T Recommendation P.800: Methods for Subjective Determination of Transmission Quality," 1996.

[25] P. Casas et al., "Exploring QoE in Cellular Networks: How Much Bandwidth do you Need for Popular Smartphone Apps?," in *SIGCOMM ATC*, 2015.

[26] T. Hoßfeld et al., "On Additive and Multiplicative QoS-QoE Models for Multiple QoS Parameters," in *PQS*, 2016.

[27] T. Hoßfeld et al., "Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience," in Data Traffic Monitoring and Analysis, 2013.