Demo: a wrapper for automated measurements with YouTube's native app

Frank Loh, Theodoros Karagkioules, Michael Seufert, Bernd Zeidler, Dimitrios Tsilimantos, Phuoc Tran-Gia, Stefan Valentin, Florian Wamser

Demo: A Wrapper for Automated Measurements with YouTube's Native App

Frank Loh*, Theodoros Karagkioules†, Michael Seufert*, Bernd Zeidler*, Dimitrios Tsilimantos†,
Phuoc Tran-Gia*, Stefan Valentin†, Florian Wamser*

*University of Würzburg, Germany, {firstname.lastname}@informatik.uni-wuerzburg.de
†Paris Research Center, Huawei Technologies France SASU, {firstname.lastname}@huawei.com

Abstract—**This demo introduces a wrapper used for automated measurements of mobile video streaming in the Android YouTube app. The difference to traditional measurement techniques is that the measurement is done with the native YouTube app as it is provided in the Google Play Store. In addition to bandwidth or packet loss detection, the QoE of the video stream can be measured and quantified. For this, the amount of quality changes, the current playtime, the buffer level, and statistics like video and audio format are captured. Thus, detailed relationships between network parameters and streaming behavior based on many factors can be detected within the native app available in the Play Store.**

I. Introduction and Differentiation

More recently, measurement methods for YouTube such as YoMo [1], YouSlow [2], YOUQMON [3], and YoMoApp [4], [5] have received much attention in order to assess quality-of-experience (QoE)-relevant features for YouTube video streaming. With more than one billion users and one billion hours watched daily, YouTube is one of the most popular services of today's Internet [6].

There are two ways to quantify YouTube streaming quality stated in literature. (1) Users' streaming QoE is estimated from network traffic only [3]. (2) In contrast, YouTube streaming quality and resulting user QoE can be quantified at the user side using a separate monitoring application [5]. This demonstration introduces another novel, different measurement methodology, listed below.

Measurement at the end user: Exact overall information is very useful for optimizing and monitoring mobile networks for specific applications. However, the widespread adoption of end-to-end encryption and the lack of common standards has made it very difficult to obtain such information by Deep Packet Inspection (DPI) or cross-layer signaling. Thus, a measurement on the user side at the app is essential to achieve reliable results. Our measurement approach is performed at the user side at regular smartphones or tablets. QoE-related information is extracted.
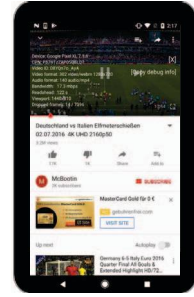

Fig. 1. YouTube App during Measurement

Measurement of the native app: We measure the native YouTube app exactly as it is delivered in the Play Store by Google. This ensures that exactly the features that belong to the current version are measured. Other approaches [1], [5] require extra apps to take a measurement. Various test measurements during the development of this approach have shown that the streaming behavior might change with app, mobile operating system, or browser version. Thus, the measurement with the native app at a regular operating system is essential. It is the first approach that measures the native app from the Play Store instead of imitating the app with content from the YouTube website.

Automated measurement procedure: We provide a complete framework for performing automated YouTube measurements. Our app emits user input and launches random or selected videos for large scale studies.

Currently, three stakeholders are particularly interested in YouTube measurements, shown in this demonstration. First, from *service provider's point of view*, information about QoE and actual streaming quality is essential to evaluate and optimize the service. Second, from *network operator's point of view*, knowledge of streaming is important to cope with the high volume of traffic from YouTube. Finally, the third party interested in YouTube measurements are *regulators*, who evaluate networks to monitor regulatory compliance for governments.
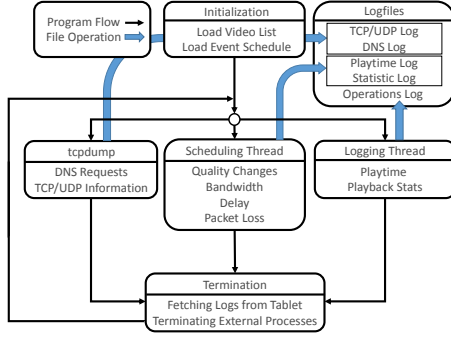
Fig. 2. Demonstration Program Flow

TABLE I
YOUTUBE AND QOE-RELATED INFORMATION FROM APPROACH

as Key	Presumable meaning	Example
csdk	Android SDK version	"csdk":"25"
c	Platform type	"c":"android"
cbrand	Phone brand	"cbrand":"Google"
cbrver	App version	"cbrver":"12.34.55"
cplayer	Player type	"cplayer":"ANDROID_EXOPLAYER"
cplatform	Player platform	"cplatform":"mobile"
cmodel	Phone model	"cmodel":"Pixel XL"
cver	App version	"cver":"12.34.55"
cbr	App package name	"cbr":"com.google.android.youtube"
cosver	OS version	"cosver":"7.1.1"
cos	OS name	"cos":"Android"
videoid	Video id	"videoid":"N2sCbtodGMI"
fmt	Video format id (itag)	"fmt":"244"
afmt	Audio format id (itag)	"afmt":"140"
bh	Buffered playtime	"bh":34900
bwe	Bandwidth estimate	"bwe":4740424

In this demonstration, QoE-related measurements of application and network parameters are performed in the native YouTube app using a wrapper application that mimics user input and extracts QoE-related information. By detecting in addition to bandwidth and packet loss QoE related parameters like streaming quality, quality changes, video, and audio format, streaming QoE is directly determined within this application.

The authors demonstrate the approach by starting a video stream with the native YouTube app on a tablet. The wrapper app is initialized to capture streaming information. With changing network conditions, the network parameters like available bandwidth are logged together with video information. The resulting logs are displayed on a separate screen for visualization.

II. IMPLEMENTATION AND REQUIRED COMPONENTS

The demonstration begins with a user starting a YouTube video stream. Subsequently, the network is controlled and network conditions are measured. Finally, the streaming behavior in the native YouTube app is captured. The complete program flow is depicted in Figure 2. The black arrows indicate the program flow, the blue arrows the file operations. First, a predefined video list and scheduled event list is loaded. Then, the program starts tcpdump, the scheduling thread, and the logging thread. By starting tcpdump, DNS requests and TCP/UDP/QUIC information is captured and written in a file. In the scheduling thread, events like quality changes, current bandwidth settings, delay, or packet loss are scheduled and monitored and written in a logfile for later evaluation. In the logging thread the current playtime and playback statistics like buffer filling are captured and stored. After the playback, tcpdump and the logging are terminated. The result files are afterwards sent from the tablet to the dashboard for evaluation and display.

A typical measurement presentation is shown in Figure 1. As depicted, the measurement can be done with the native app. Thus, the measure does not affect the user's YouTube watching behavior. Additionally, the *stats for nerds* overlay and the timestamp above the seek bar is displayed, which contain many streaming- and QoE-related parameters like video and audio quality or timestamp. A complete list is presented in Table I.

III. CONCLUSION

In this demo, a wrapper for automated measurements of mobile video streaming in the Android YouTube app is presented. It is a novel method to measure and analyze video streaming with YouTube's Android app. The difference to traditional measurement techniques is that the measurement is done with the native YouTube app as it is provided in the Google Play Store.

REFERENCES

[1] B. Staehle, M. Hirth, R. Pries, F. Wamser, and D. Staehle, "Yomo: A youtube application comfort monitoring tool," in *New Dimensions in the Assessment and Support of Quality of Experience for Multimedia Applications*, Tampere, Finland, 6 2010.

[2] H. Nam, K.-H. Kim, D. Calin, and H. Schulzrinne, "Youslow: a performance analysis tool for adaptive bitrate video streaming," *ACM SIGCOMM Computer communication review*, vol. 44, no. 4, pp. 111–112, 2015.

[3] P. Casas, M. Seufert, and R. Schatz, "Youqmon: A system for on-line monitoring of youtube qoe in operational 3g networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 2, pp. 44–46, 2013.

[4] F. Wamser, M. Seufert, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "Poster: Understanding youtube qoe in cellular networks with yomoapp: A qoe monitoring tool for youtube mobile," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 263–265.

[5] ——, "Yomoapp: A tool for analyzing qoe of youtube http adaptive streaming in mobile networks," in *Networks and Communications (EuCNC), 2015 European Conference on*. IEEE, 2015, pp. 239–243.

[6] "Youtube for press," https://www.youtube.com/yt/about/press/, accessed: 2018-01-15.