Towards a Framework for Classifying YouTube QoE Based on Monitoring of Encrypted Traffic

Irena Orsolic, Lea Skorin-Kapov and Mirko Suznjevic
University of Zagreb, Faculty of Electrical Engineering and Computing
10000 Zagreb, Croatia
Email: {irena.orsolic, lea.skorin-kapov, mirko.suznjevic}@fer.hr

***Abstract*—With the move to traffic encryption adopted by many Over The Top (OTT) providers of video distribution services, Internet Service Providers (ISPs) are now facing the challenges of monitoring application performance and potential end user perceived service quality degradations. With lack of direct feedback from OTT providers, ISPs generally rely on passive traffic monitoring solutions deployed within their network for the purposes of monitoring OTT service performance. In this paper we describe our ongoing research efforts aimed at investigating solutions for estimating end user QoE when watching YouTube videos, based solely on the analysis of encrypted traffic in mobile and WiFi networks. We shortly describe our developed YouQ system which enables the monitoring of both application-layer KPIs and encrypted network traffic for the purpose of developing ML-based QoE classification models. We discuss ongoing and future work in the direction of developing a more general framework for the estimation of video streaming QoE based on further enhancements of the YouQ system. The framework aims to support the collection of data across different end user device platforms and access networks, and the analysis of both TCP and QUIC traffic.**

I. Introduction

We are witnessing a constant growth in mobile data traffic, with devices getting more powerful and users increasingly embracing high-resource-demanding multimedia services on the move. Currently, according to Cisco's Visual Networking Index [1], the largest amount of mobile data traffic is video content, which makes up approximately 60% of overall mobile traffic, with the number expected to grow to 75% over the next five years.

To optimize the resource utilization and to meet users' needs and expectations in terms of both Quality of Service (QoS) and Quality of Experience (QoE), actors in the service delivery chain employ various strategies. Large Over The Top (OTT) video streaming providers rely on CDNs for delivering and caching content [2] [3]. To reduce bandwidth consumption and improve QoE, many video delivery services (e.g., YouTube and Netflix) rely on the MPEG-DASH standard [4] for dynamically adapting video delivery to varying network conditions and buffer status.

Prior to the widespread use of encryption in OTT traffic, network operators could gain insight into application performance by extracting packet header information. Today, the inability to monitor service performance at an application level poses a threat to network providers, as they are potentially unable to detect problems and act accordingly. Poor performance further imposes the risk of losing customers, as customers often tend to blame network providers for poor QoE. Given the current situation, ISPs commonly rely on passive traffic monitoring solutions deployed within their network to obtain insight into user perceived degradations and their potential causes.

To address these challenges faced by ISPs, we have been studying the feasibility of estimating YouTube QoE based on monitoring of encrypted network traffic, by using machine learning (ML) techniques. To do so, we are developing a system called **YouQ**. Developing such a system includes tackling multiple challenges. There is a need to understand how application Key Performance Indicators (KPIs) (such as stalling duration, initial delay, etc.) affect end users' QoE (QoE modelling problem). The survey given by Seufert *et al.* [5] gives a comprehensive overview of subjective studies that cover QoE aspects of adaptation, and discusses QoE influence factors and corresponding QoE models for adaptive video streaming. Although a number of QoE models have been proposed [6]–[10], very few approaches simultaneously map the impact of multiple influence factors onto QoE [11], [12]. However, until recently there was no widely accepted multidimensional QoE model in literature or standards which takes into account stalling duration/length, initial delays, and adaptation behaviour. We note that in Nov. 2016 the ITU-T (SG12) published Recommendation P.1203: "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport" [13] (formerly referred to as P.NATS). The Recommendation describes a set of objective parametric quality assessment modules that together can be used to form a complete model to predict the impact of audio and video media encodings and observed IP network impairments on quality experienced by the end-user in multimedia streaming applications.

To detect QoE degradations and identify the root causes of QoE impairments, monitoring solutions can be deployed on client devices. In [14], the authors analysed the influence of both constant and dynamically changing network access conditions to better understand the QoE requirements of popular mobile apps, including YouTube. In the case of YouTube, they used YoMoApp (YouTube Performance Monitoring Application) developed by Wamser *et al.* [15], an Android application that passively monitors application-layer YouTube KPIs. YoMoApp uses the YouTube IFrame API [16] to obtain the KPIs. We also employ this approach in the scope of our YouQ client application, but are also working on including different client applications that embed the YouTube player to cover different usage scenarios.

Another challenge lies in extracting the network traffic features that can be correlated to application-level degradations. Dimopoulos *et al.* [17] proposed a methodology for detecting video streaming QoE issues from encrypted traffic. The changes in size and inter-arrival times of video segments proved to be among the most important indicators of quality impairments. Similarly, Shafiq in [18] describes a methodology for network-side video QoE measurement and monitoring in mobile networks that works with encrypted traffic and can predict video QoE by observing only the initial 10 seconds of a video streaming session. Aggarwal *et al.* proposed the Prometheus system [19], which relies on ML techniques to relate passive in-network measurements to application's QoE. It measures QoE of video-on-demand and VoIP applications without requiring knowledge about specific application services. Another study that aims at predicting QoE from network-level measurements, but focuses on the Skype use case was described in [20]. The authors consider various ML classifiers to classify sessions into four classes (Good, Medium, Poor, No Call) and report an average prediction accuracy of 66%.

Traffic feature extraction becomes much more complicated when various use cases are considered, including different client devices, with different operating systems, running different types of YouTube players, and connected to different types of network. All of these parameters affect YouTube's adaptation algorithm, and thus QoE prediction models relying on traffic features may be applicable only for certain scenarios. A number of studies have analyzed the characteristics of YouTube traffic [21], [22], and client-side behavior [23]–[26]. However, as the YouTube service changes its adaptation logic and traffic patterns (which have been found to vary across different platforms and access networks), there is a need to continuously update findings related to traffic characterization.

The idea behind the YouQ system is to enable data collection, processing and model building under a variety of conditions. It is important for the entire process to be automated, so as to simplify the model building when YouTube deploys changes in its adaptation logic. The initial YouQ system which we developed, described in Section II, proved to provide very promising results in estimating QoE for cases when YouTube was accessed via a browser on a smartphone, and over a Wi-Fi network, with TCP as the underlying transport protocol [27], [28]. However, there is a need to address other scenarios as well, including cases when YouTube is accessed via the YouTube App (in which case we observe QUIC as the underlying transport protocol), cases when delivery is over a mobile network, and cases involving various types of end user interactions (e.g., browsing videos, seeking forward/backward, etc.) [29]. These challenges and our ongoing work are discussed in Section III, with the overall aim being to work towards a framework for the estimation of video streaming QoE based on monitoring of encrypted network traffic.

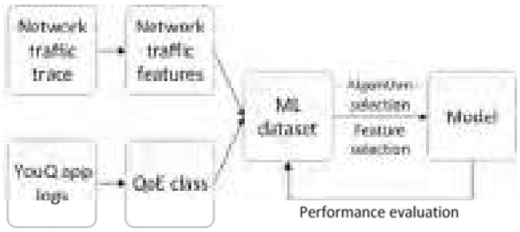

Fig. 1: Approach for QoE classification based on network traffic features

II. Methodology and current results

Our developed YouQ system consists of an Android application that plays a requested number of YouTube videos and logs events at an application level (video playing, buffering, pause, quality switch, video ended), amount of video in the buffer, and URLs from all HTTP requests. This data is collected using the YouTube IFrame API (further referred to as ***YouQ IF***). In parallel to logging of application-layer KPIs, network traffic is also captured. After all the videos are played, both application-level logs and network traffic are uploaded to a **YouQ server** and processed. Processing includes calculating traffic features (e.g., average throughput, average interarrival time, etc.) for each of the videos in the experiment, calculating application-level KPIs from the collected logs (e.g., percentage of time spent on each quality level, stalling duration, initial delay), and assigning a *QoE class* ("high", "medium", "low" QoE) to each video according to calculated KPIs and QoE models defined in [28]. The output of this phase is a dataset for training ML models. For each viewed video, we extract 33 traffic features (listed in [28]) based on the analysis of encrypted traffic, and classify the video into one of the three aforementioned QoE classes. The approach is depicted in Figure 1.

Our approach was tested in a laboratory environment shown in Figure 2. YouTube traffic between an Android smartphone (Samsung S6 with Android version 5.1.1) and YouTube servers is transmitted over an IEEE 802.11n wireless network and then routed through a PC running IMUNES (www.imunes.net), a general purpose IP network emulation/simulation tool enabling a test administrator to set up different bandwidth limitations and schedule bandwidth changes [30]. Traffic is further sent through Albedo's Net.Shark device (a portable network tap used for aggregating and mirroring network traffic) where it is replicated and sent to a PC designated for network traffic capturing. The PC running IMUNES also has an OS layer, accessed by the YouQ client application to run a bandwidth scheduling script according to defined experiments. The script resets the bandwidth envelope for each video in the experiment, which enables all videos to be played under exactly the same network conditions.

We collected a dataset corresponding to 1060 videos played under 39 differently defined bandwidth conditions, and trained models by using various ML algorithms. The models proved

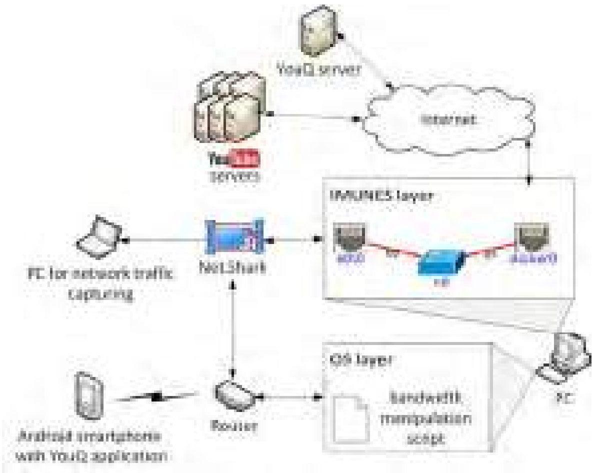

Fig. 2: YouQ lab setup

to be up to 84% accurate when 3 QoE classes were defined ("high", "medium", "low"). We also classified videos into 2 QoE classes ("high" and "low") and repeated the procedure. These models achieved an accuracy of 91%. The exact measurement procedure, definition of QoE classes, statistics of the collected dataset, along with a more detailed view of the results were published in [27], while [28] gives an even more detailed view of the test methodology used to train ML models for QoE classification, and provides a more detailed interpretation of classification results.

III. Ongoing activities and challenges

Our current activities aim to further develop the YouQ system so as to enable data collection and the training of QoE classification models for different usage scenarios. By this we refer to scenarios in which YouTube is delivered over different access networks (WiFi and mobile), using different clients (browser-based vs. YouTube App), using different transport protocols (TCP, QUIC), and with different types of user behaviour observed. Introducing this kind of variety has implications on the processing part of the system. The idea is to split the processing features on the YouQ server into simple modules that can be applied depending on the use case.

A. Different client applications

Besides the YouTube IFrame API, YouTube also offers a native Android API [31]. The Android API does not include methods for checking the quality level of a played video, but only notifies when a video starts buffering, playing, and when a video ends. We have developed a version of the YouQ Android application based on this API, which enables us to observe YouTube KPIs and traffic behaviour in the case when a user access YouTube via the dedicated YouTube App (in the remaining text referred to as ***YouQ AA***). When we compared the measurements obtained from YouQ AA (case when YouTube is accessed from the native App) and YouQ IF (case when YouTube is accessed from a browser), we see that while for YouQ IF content is sent using TCP/TLS, YouQ AA uses QUIC [32], a transport-layer protocol initially proposed by Google, and designed to reduce connection and transport latency as compared to TCP.

Our further analysis showed differences in YouTube's adaptation algorithm in cases when collecting measurements using YouQ IF and YouQ AA. During the period of January and February 2017, we ran experiments in which we played a series of 10 predefined videos under 7 different bandwidth conditions (Table I), using both YouQ IF and YouQ AA. The arrows in the table indicate the time after which available bandwidth was changed from the first value to the second value. Five out of seven bandwidth envelopes were chosen from the set of 39 envelopes we used in [27], as these resulted in higher degradations (high stalling count, high initial delay, etc.). We chose 5 Mbps (Exp. No. 1) to see the difference between YouQ IF and YouQ AA when the bandwidth is just high enough for YouTube to play 1080p videos, while in the last experiment we wanted to see the difference when the bandwidth was very high (more than 100 Mbps). Note that the videos are differing in content type and view count and were not evaluated subjectively.

TABLE I: Bandwidth envelopes used in experiments we ran to compare YouQ Android applications based on YouTube IFrame API (YouQ IF) and Android API (YouQ AA)

Experiment No.	Bandwidth [Mbps]
1	5
2	0.4
3	$0.5 \xrightarrow{60s} 3$
4	$0.5 \xrightarrow{60s} 10 \xrightarrow{40s} 0.5$
5	$0.85 \xrightarrow{60s} 0.75$
6	$10 \xrightarrow{60s} 0.5$
7	more than 100

We observed that in 6 out of 7 network scenarios, .pcap files captured during playing all 10 videos (in a playlist) were larger in the case of using YouQ AA (Figure 3a) (note that traffic size in each experiment corresponds to delivery of the same 10 predefined videos). By observing these sizes, we can assume that the YouTube Android App player is more aggressive than YouTube web player. In other words, we assume this was due to the App downloading higher video quality levels under the same network conditions. In experiments we conducted earlier and described in [27], we noticed that YouQ IF plays videos in quality level *large* (480p) even when the bandwidth is high and quality level *hd1080* (1080p) is available. With YouQ AA this may not be the case. This is especially noticeable in experiment No. 7, where no bandwidth limitations were set and measured bandwidth was higher than 100 Mbps.

Traffic size might also vary because of the differences in video codec. As methods that notify of quality switches are only available in the IFrame API and not using the Android API, we are currently exploring the possibility of logging quality levels in YouQ AA in other ways. Having this information would provide us more detailed insights into YouTube's adaptation behaviour and application layer KPIs, needed to train ML models in the YouQ AA case.

(a) Size of captured traffic in an experiment

(b) Average initial delay in an experiment

(c) Number of stallings throughout the experiment

(d) Average stalling duration in an experiment

Fig. 3: Differences in behaviour of YouQ Android applications based on YouTube IFrame API (YouQ IF) and Android API (YouQ AA). Each experiment is defined by type of application (IF/AA) and bandwidth envelope given in Table I. Throughout each experiment the list of 10 predefined videos was played.

As the Android API player requests a higher amount of traffic, assumed to correspond to higher quality video, YouQ AA experiments on low bandwidths resulted in higher stalling count and longer stalling durations (Figure 3c and Figure 3d), despite the fact that YouQ AA uses QUIC as opposed to TCP/TLS. Initial delays were longer in all of the experiments except experiments No. 6 and 7 (Figure 3b).

B. Different transport protocols

Using as a client device Samsung S6, we found that in all access network cases (WiFi, 3G, 4G) and using both the Chrome browser (version 55.0.2883.91) and the YouTube app (version 12.01.55), QUIC was used as the default protocol (Feb. 2017). Based on previous measurements and literature review, it is evident that usage of the QUIC protocol can be (and is) enabled and disabled as Google studies the effects of introducing this protocol. Google in [33] reported improvements in terms of performance over TCP, due to QUIC's lower-latency connection establishment, improved congestion control, and better loss recovery. They also stated the intention to make QUIC the default transport from Google clients (Chrome and mobile apps) to Google servers, and to formally propose QUIC to the IETF as an Internet standard, which has already happened [32]. Nevertheless, it seems highly likely that QUIC will in the future be the base for YouTube functionality. Therefore, we plan to do an in depth study of the QUIC protocol and its impact on performance of the YouTube service both on application and network level. As YouQ currently extracts network traffic features from TCP traffic only, to enable building models that work with QUIC traffic, we plan on defining a set of QUIC traffic features that can be correlated to QoE, and incorporating the calculation of these features into YouQ.

Furthermore, new methodologies can be applied for improvement of the existing TCP traffic feature extraction. The authors in [34] propose a machine learning based bitrate estimation approach to parse bitrate information from IP packet level measurements, with a focus on encrypted adaptive YouTube video streaming. Incorporation of the estimated video bitrate/quality level as an additional traffic feature provides the potential to improve the accuracy of previously developed ML-based QoE classification models.

C. User behaviour

Finally, we are investigating different aspects of user behaviour and their effects on the service. We plan on running experiments and collecting training data involving different user interactions (e.g., using a playlist, autoplay, manually browsing through videos, seeking forward/backward, etc.), to determine the implications with respect to developing ML-based QoE classification models.

IV. Conclusion

With the encryption of OTT traffic, network providers commonly lack insight into application performance. Given the complex relationships between network performance and user perceived quality, estimating QoE based solely on encrypted network traffic passing through network is a challenging problem.

We have addressed this challenge by considering the case of YouTube and developing a system called YouQ that uses machine learning techniques for analysing both application and network level data, to create a model that can estimate YouTube video streaming QoE based on traffic characteristics. To apply the described methodology in a real operational network, the experiments would need to be run to collect data on YouQ clients and, in parallel, capture traffic on network probes. In such a test setup, it would further be possible to address the impact of real network fluctuations.

This paper described the developed system in short and presented challenges and possible solutions on the way to extend the functionalities of the system. We introduced the newly developed client applications and provided an overview of use cases we aim to consider, aimed to develop a generalized framework for the estimation of YouTube QoE from encrypted traffic. In future work we will also explore the potential of applying the system (methodology and tools) to other adaptive video streaming services beyond YouTube.

Acknowledgment

This research is conducted in the scope of projects "Survey and analysis of monitoring solutions for YouTube network traffic and application layer KPIs" and "QoMoVid: QoE Monitoring Solutions for Mobile OTT Video Streaming", both funded by Ericsson Nikola Tesla, Croatia. This work has also been supported in part by the Croatian Science Foundation under the project UIP-2014-09-5605 (Q-MANIC). The authors would like to thank Illona Popic, Petra Rebernjak and Ivan Bartolec for their help in running experiments.

REFERENCES

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20162021," Cisco, Tech. Rep., 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[2] T. Böttger, F. Cuadrado, G. Tyson, I. Castro, and S. Uhlig, "Open Connect Everywhere: A Glimpse at the Internet Ecosystem through the Lens of the Netflix CDN," *arXiv preprint arXiv:1606.05519*, 2016.

[3] P. Casas, A. D'Alconzo, P. Fiadino, A. Bar, A. Finamore, and T. Zseby, "When YouTube Does not Work-Analysis of QoE-Relevant Degradation in Google CDN Traffic," *Network and Service Management, IEEE Transactions on*, vol. 11, no. 4, pp. 441–457, 2014.

[4] I. Sodagar, "MPEG-DASH: The Standard for Multimedia Streaming Over Internet," International organisation for standardisation, WA, USA 98052, White paper ISO/IEC JTC1/SC29/WG11 W13533, apr 2012.

[5] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[6] 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.

[7] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau, "Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience," in *Data Traffic Monitoring and Analysis*. Springer, 2013, pp. 264–301.

[8] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea," in *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*. IEEE, 2012, pp. 1–6.

[9] D. Ghadiyaram, A. C. Bovik, H. Yeganeh, R. Kordasiewicz, and M. Gallant, "Study of the Effects of Stalling Events on the Quality of Experience of Mobile Streaming Videos," in *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*. IEEE, 2014, pp. 989–993.

[10] T. Hoßfeld, M. Seufert, C. Sieber, and T. Zinner, "Assessing Effect Sizes of Influence Factors Towards a QoE Model for HTTP Adaptive Streaming," in *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*. IEEE, 2014, pp. 111–116.

[11] J. Xue, D.-Q. Zhang, H. Yu, and C. W. Chen, "Assessing Quality of Experience for Adaptive HTTP Video Streaming," in *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–6.

[12] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao, "Deriving and Validating User Experience Model for DASH Video Streaming," *IEEE Transactions on Broadcasting*, vol. 61, no. 4, pp. 651–665, 2015.

[13] T. standarization sector of ITU, "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport," International Telecommunication Union, Tech. Rep. P.1203, nov 2016.

[14] P. Casas, M. Seufert, F. Wamser, B. Gardlo, A. Sackl, and R. Schatz, "Next to You: Monitoring Quality of Experience in Cellular Networks from the End-devices," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 181–196, 2016.

[15] F. Wamser, M. Seufert, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "YoMoApp: A Tool for Analyzing QoE of YouTube HTTP Adaptive Streaming in Mobile Networks," in *Proc. of EuCNC 2015*. IEEE, 2015, pp. 239–243.

[16] "YouTube Player API Reference for iframe Embeds." [Online]. Available: https://developers.google.com/youtube/iframe_api_reference

[17] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki, "Measuring Video QoE from Encrypted Traffic," in *Proceedings of the 2016 ACM on Internet Measurement Conference*. ACM, 2016, pp. 513–526.

[18] M. Z. Shafiq, "Tracking Mobile Video QoE in the Encrypted Internet," *White-paper submission, U. of Iowa*, p. 3, 2015.

[19] V. Aggarwal, E. Halepovic, J. Pang, S. Venkataraman, and H. Yan, "Prometheus: Toward Quality of Experience Estimation for Mobile Apps from Passive Network Measurements," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. ACM, 2014, p. 18.

[20] T. Spetebroot, S. Afra, N. Aguilera, D. Saucez, and C. Barakat, "From network-level measurements to expected quality of experience: the skype use case," in *Measurements & Networking (M&N), 2015 IEEE International Workshop on*. IEEE, 2015, pp. 1–6.

[21] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "Youtube Everywhere: Impact of Device and Infrastructure Synergies on User Experience," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 345–360.

[22] J. J. Ramos-Muñoz, J. Prados-Garzon, P. Ameigeiras, J. Navarro-Ortiz, and J. M. López-Soler, "Characteristics of Mobile YouTube Traffic," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 18–25, 2014.

[23] A. Mansy, M. Ammar, J. Chandrashekar, and A. Sheth, "Characterizing Client Behavior of Commercial Mobile Video Streaming Services," in *Proceedings of Workshop on Mobile Video Delivery*. ACM, 2014, p. 8.

[24] C. Sieber, A. Blenk, M. Hinteregger, and W. Kellerer, "The Cost of Aggressive HTTP Adaptive Streaming: Quantifying YouTube's Redundant Traffic," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1261–1267.

[25] G. Horvath and P. Fazekas, "Modelling of YouTube Traffic in High Speed Mobile Networks," in *European Wireless 2015; 21th European Wireless Conference; Proceedings of*. VDE, 2015, pp. 1–6.

[26] F. Wamser, P. Casas, M. Seufert, C. Moldovan, P. Tran-Gia, and T. Hossfeld, "Modeling the YouTube Stack: From Packets to Quality of Experience," *Computer Networks*, 2016.

[27] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, "YouTube QoE Estimation Based on the Analysis of Encrypted Network Traffic Using Machine Learning," in *Globecom Workshops (GC Wkshps), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[28] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, "A Machine Learning Approach to Classifying YouTube QoE Based on Encrypted Network Traffic," *Multimedia tools and applications*, 2017, doi:10.1007/s11042-017-4728-4.

[29] T. Hoßfeld, C. Moldovan, and C. Schwartz, "To Each According to his Needs: Dimensioning Video Buffer for Specific User Profiles and Behavior," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015, pp. 1249–1254.

[30] M. Zec and M. Mikuc, "Operating System Support for Integrated Network Emulation in IMUNES," in *Workshop on Operating System and Architectural Support for the on demand IT Infrastructure (1; 2004)*, 2004.

[31] "Google APIs for Android." [Online]. Available: https://developers.google.com/android/guides/api-client

[32] R. Hamilton, J. Iyengar, I. Swett, and A. Wilk, "QUIC: A UDP-based secure and reliable transport for HTTP/2," *IETF, draft-tsvwg-quic-protocol-02*, 2016.

[33] "A QUIC update on Googles experimental transport." [Online]. Available: https://blog.chromium.org/2015/04/a-quic-update-on-googles-experimental.html

[34] W. Pan, G. Cheng, H. Wu, and Y. Tang, "Towards QoE Assessment of Encrypted YouTube Adaptive Video Streaming in Mobile Networks," in *Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on*. IEEE, 2016, pp. 1–6.