Real-Time QoE estimation for DASH video using active network probing

Reference:
Miranda Gilson, Municio Esteban, Marquez-Barja Johann, Macedo Daniel Fernandes.- Real-Time QoE estimation for DASH video using active network probing
Intelligence in Next Generation Networks (ICIN), International Conference on - ISSN 2162-3414 - 2022, p. 157-159
Full text (Publisher's DOI): https://doi.org/10.1109/ICIN53892.2022.9758135
To cite this reference: https://hdl.handle.net/10067/1876750151162165141

Real-Time QoE Estimation for DASH Video Using Active Network Probing

Gilson Miranda Jr.*†, Esteban Municio*, Johann M. Marquez-Barja*, Daniel Fernandes Macedo†
*University of Antwerp - imec, IDLab, Faculty of Applied Engineering - Antwerp, Belgium
†Universidade Federal de Minas Gerais - Computer Science Department - Minas Gerais, Brazil
E-mail: {gilson.miranda, esteban.municio, johann.marquez-barja}@uantwerpen.be, damacedo@dcc.ufmg.br

***Abstract*—Video on Demand (VoD) accounts for a significant amount of traffic on IP networks. To meet users' expectations, network operators need means to monitor and to identify when service quality is degraded in order to take actions to avoid customer churn. Most solutions cannot monitor end-to-end conditions without modification on video player applications or require deep packet inspection techniques, which may raise privacy issues. In this demonstration, we use active network probing to measure end-to-end network Quality of Service (QoS) conditions and use a Machine Learning model to infer users' Quality of Experience (QoE) in real-time. The results show that the method allows us to identify whether the network conditions allow video sessions with high QoE, or situations in which the user's QoE is degraded.**

***Index Terms*—DASH Video, QoE, Machine Learning**

I. Introduction

Video on Demand (VoD) accounts for large amounts of traffic over the Internet. To keep users satisfied with their experience, content and network operators must be able to identify when user experiences are unsatisfactory. With the widespread consumption of VoD services such as Netflix and YouTube, industry and academia have been seeking ways to monitor user-perceived quality for such services [1]. However, network Quality of Service (QoS) metrics like bandwidth, delays, or Packet Loss Ratio (PLR) do not map directly into the user-perceived experience [2]. Instead, the concept of Quality of Experience (QoE) is used to measure or estimate the user's subjective perception of a service.

Many proposals in the literature lack the ability to monitor the last-mile link, which in many cases is the network bottleneck, especially in wireless networks. Therefore, QoE monitoring methods that do not cover the last mile may be unable to detect QoE issues. In previous work, we proposed a method to monitor QoE for VoD using Internet Control Message Protocol (ICMP) probing and a Machine Learning (ML) model that takes network QoS as input and estimates QoE [3]. In this work, We demonstrate the effectiveness of the method through real-time monitoring using a setup comprised by the CityLab[1] [4] smart city/wireless testbed, and Virtual Wall[2] cloud testbed. We use an improved version of the inference model with more relevant QoS statistics, and a secondary model that improves inference accuracy.

[1]https://doc.lab.cityofthings.eu/wiki/Main_Page
[2]https://doc.ilabt.imec.be/ilabt/virtualwall/

II. QoE Inference Using ICMP Probing

Our method uses ICMP probing to perform end-to-end network QoS measurements. Figure 1 gives an overview of the method in co-located and distributed deployments. The co-location considers a context of small-scale Content Delivery Networks (CDNs) deployed within the domain of an Internet Service Provider (ISP). The server provides VoD based on Dynamic Adaptive Streaming over HTTP (DASH). The ISP has no access to server logs but can deploy a Probing Module (PM) to monitor the network between server and client. The ISP can also configure routes so the probing and video flows follow the same path. In the distributed context the PM is deployed in a network point between the server and the client, performing independent probing on both hosts. The measurements are aggregated to obtain the end-to-end QoS and perform MOS inferences.

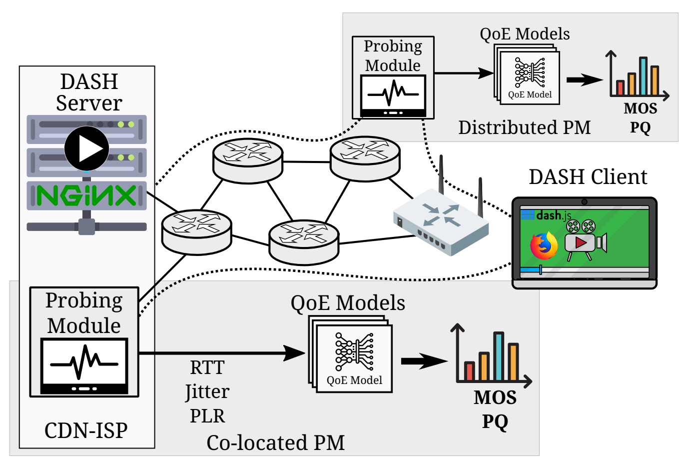

Fig. 1. Overview of the QoE inference method

The PM continuously measures the Round-Trip Time (RTT), jitter, and PLR between server and client, running parallel probing threads. Probing intervals are adjusted according to the RTT to obtain 1000 samples in 30 seconds. RTT, jitter, and PLR statistics are given as input to ML models based on eXtreme Gradient Boosting (XGBoost) [5]. The models map the input QoS information into an Mean Opinion Score (MOS) value between 1 and 5, based on ITU-T P.1203 Recommendation. The dataset used to train the models was created using the emulated setup in Figure 2. The server offers a catalog of 15 videos obtained from *4kmedia.org*, each encoded in 10 quality levels. Network impairments between DASH Server, PM, and DASH Client were inserted using the Traffic Control (TC) tool for Linux. We executed over 114.000 video sessions and used the software provided by Robitza et al. and Raake et al. [6], [7]. to label the MOS of the sessions at each second. The DASH client was based on the DASH Industry Forum reference player v4.0.0.

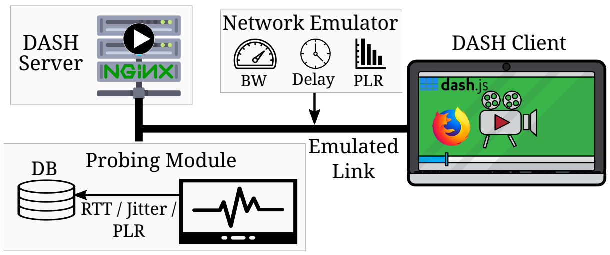

Fig. 2. Emulated setup for dataset creation and model training

The inference method is shown in Figure 3. The models are based on supervised learning, using a regression tree ensemble built using XGBoost[3]. The PM gathers median RTT, 90th percentile of RTT, median jitter, 90th percentile of jitter, and PLR from the last 30 seconds. The *Primary Model* estimates MOS each second. Per-second MOS estimates are accumulated by the *Postprocess* module, which calculates six statistics: standard deviation of MOS values for the last 10, 20, and 30 seconds; and mean MOS values for the last 10, 20, and 30 seconds. The *Secondary Model* gets QoS statistics from the PM, the most recent MOS estimated by the Primary Model (MOS Pass 1), and the statistics from the Postprocess module. The output of the Secondary Model is the final MOS inference. We performed over 115,000 video sessions to build a dataset for model training and used 20 % of the sessions to evaluate the model, achieving an Root Mean Square Error (RMSE) of 1.04. In this demonstration we show the use of the model on real deployments.

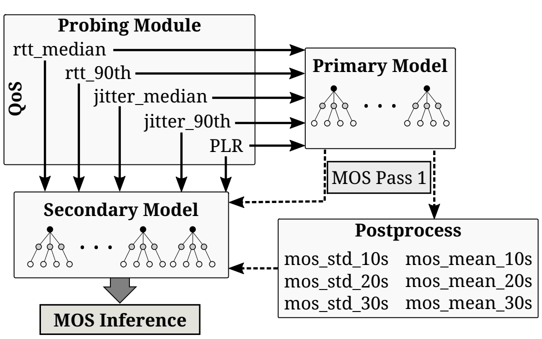

Fig. 3. Overview of the inference model

III. Demonstration Scenario

For this demonstration, we deploy the video server on Virtual Wall nodes, and use pairs of CityLab nodes as Wi-Fi APs and clients. The nodes used on Virtual Wall are *pcgen2* with 2 Quad-core Intel E5520 2.2GHz CPUs and 12GB of RAM. On CityLab the nodes are PC Engines apu2c4[4] with an AMD GX-412TC 1GHz Quad-core CPU, 4GB of RAM, and Atheros QCA9880 Wi-Fi cards. The setups to be part of the demonstration are listed below (Figure 4 shows the position of the outdoors setups):

- **Setup 1:** nodes 6 (AP) and 72 (Client). Indoors, with nodes in close proximity. 40 other APs in range, 6 on the same channel.
- **Setup 2:** nodes 71 (AP) and 6 (Client). Indoors, with nodes in close proximity. 37 APs in range of the AP, 9 on the same channel. 41 APs in range of the client, 6 on the same channel.
- **Setup 3:** nodes 24 (AP) and 28 (Client). Outdoors, with 50 APs in range of the AP, 1 on the same channel. 20 APs in range of the client, 5 on the same channel.
- **Setup 4:** nodes 14 (AP) and 18 (Client). Outdoors. 107 APs in range of the AP, 10 on the same channel. 39 APs in range of the client, 4 on the same channel.
- **Setup 5:** nodes 34 (AP) and 35 (Client). Outdoors. 37 APs in range of the AP, 7 on the same channel. 144 APs in range of the client, 27 on the same channel.

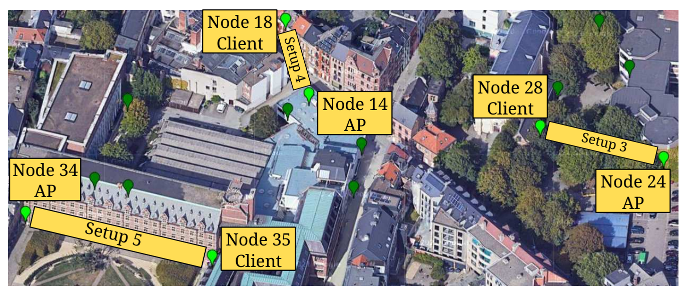

Fig. 4. Setups 3 to 5 using outdoors nodes of CityLab

The VoD server and the PM are deployed on separate containers on Virtual Wall. For the demonstration, the PM performs QoS monitoring, and also runs the inference model to obtain real-time QoE estimates. The statistics can be viewed in real-time in a Dashboard using Grafana, including the QoS values probed by the PM, the QoE estimated, and the video quality level being played. The client is instrumented to show the video quality level being consumed during the session, as well as other application-level statistics. During the demonstration, the user can visualize such monitoring information, as well as the QoE level estimated by our method.

IV. Demonstration Output

Figures 5, 6, and 7 show examples of the inference outputs values seen in the Dashboard. Figure 5 shows a session of the "jimix" video on setup 1. The client constantly receives high MOS, and we can observe that the inferences are close to the measurements with slight oscillations.

[3]https://xgboost.ai

[4]https://pcengines.ch/apu2c4.htm

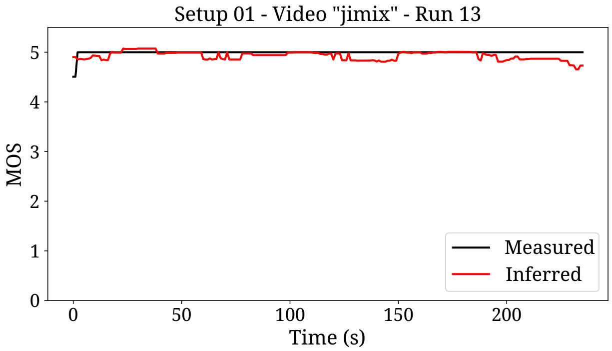

Fig. 5. Sample session on setup 1 and video ”jimix”.

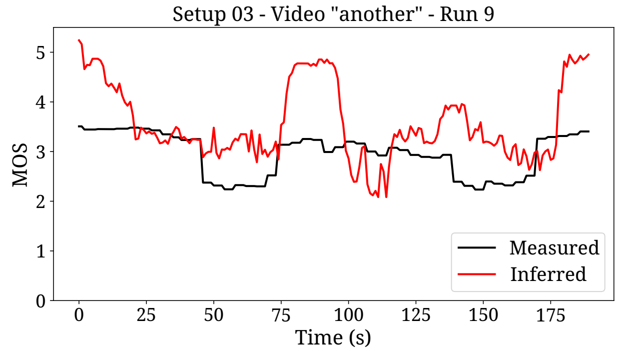

Fig. 7. One video session of the ”another” video on setup 3.

Figure 6 shows a session of the “travel” video on setup 2. In that case, the inferred values oscillate more, and in fact, for a period of 25 seconds at the beginning of the session there was a drop in MOS. This indicates that the network conditions could be improved in order to guarantee the highest possible QoE for the whole session. On setup 3 the sessions had more oscillations, as shown in the example of Figure 7 (a session of the “another” video on setup 3), and also on the standard deviation of mean inferred MOS values. We observe that during some periods in Figure 7 the inferred MOS values present higher errors. Nevertheless, for most of the session duration, the error is within the expected RMSE, and the oscillation level of inferences can also be used as an indicator of sub-optimal user experience.

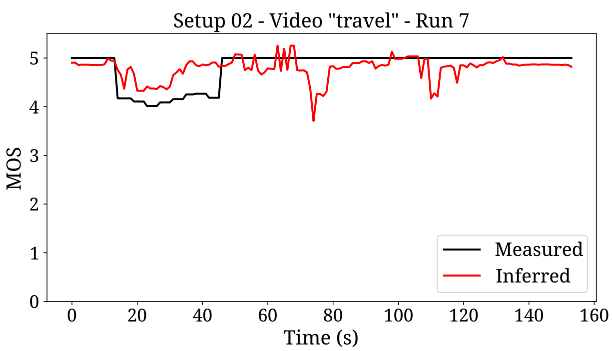

Fig. 6. Sample session on setup 2 and video ”travel”.

For the demonstration, the client is instrumented to send playback statistics to a controller node, using a control link of the testbed. The controller also aggregates the latest data and runs the ITU-T P.1203 models to obtain the “Measured” MOS, while our inference method provides the “Inferred” MOS.

V. Conclusion

In this work, we demonstrate our method of QoE inference for DASH video using active network probing and ML. The experimental results using realistic wireless setups show the feasibility of the method and its limitations. The demonstration allows real-time visualization of the MOS calculated using ITU-T P.1203 models, and the inferred values inferred by our method using network-level statistics.

Acknowledgment

This work was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, CNPq (funding agency from the Brazilian federal government), FAPEMIG (Minas Gerais State Funding Agency), and São Paulo Research Foundation (FAPESP) with Brazilian Internet Steering Committee (CGI.br), grants 2018/23097-3 and 2020/05182-3.

The work has also been supported by the Horizon 2020 projects Fed4FIRE+ (Grant Agreement No. 723638), 5G-Blueprint (Grant Agreement No. 952189), and by the FLEXNET project: “Flexible IoT Networks for Value Creators” (Celtic 2016/3), in the Eureka Celtic-Next Cluster.

References

[1] M. Yang, S. Wang, R. N. Calheiros, and F. Yang, “Survey on QoE assessment approach for network service,” *IEEE Access*, vol. 6, pp. 48 374–48 390, 2018. [Online]. Available: doi.org/10.1109/ACCESS.2018.2867253

[2] M. Fiedler, T. Hossfeld, and P. Tran-Gia, “A generic quantitative relationship between quality of experience and quality of service,” *IEEE Network*, vol. 24, no. 2, pp. 36–41, 2010. [Online]. Available: doi.org/10.1109/MNET.2010.5430142

[3] G. Miranda, D. F. Macedo, and J. M. Marquez-Barja, “A QoE Inference Method for DASH Video Using ICMP Probing,” in *2020 16th International Conference on Network and Service Management (CNSM)*, 2020, pp. 1–5. [Online]. Available: doi.org/10.23919/CNSM50824.2020.9269120

[4] J. Struye, B. Braem, S. Latré, and J. Marquez-Barja, “The CityLab testbed—Large-scale multi-technology wireless experimentation in a city environment: Neural network-based interference prediction in a Smart City,” in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 529–534. [Online]. Available: doi.org/10.1109/INFCOMW.2018.8407018

[5] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016. [Online]. Available: doi.org/10.1145/2939672.2939785

[6] A. Raake, M.-N. Garcia, W. Robitza, P. List, S. Göring, and B. Feiten, “A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P.1203.1,” in *International Conference on Quality of Multimedia Experience (QoMEX)*, May 2017. [Online]. Available: doi.org/10.1109/QoMEX.2017.7965631

[7] W. Robitza, S. Göring, A. Raake, D. Lindegren, G. Heikkilä, J. Gustafsson, P. List, B. Feiten, U. Wüstenhagen, M.-N. Garcia, K. Yamagishi, and S. Broom, “HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P.1203 – Open Databases and Software,” in *ACM Multimedia Systems Conference*, 2018. [Online]. Available: doi.org/10.1145/3204949.3208124