

A QoE Inference Method for DASH Video Using ICMP Probing

Gilson Miranda Jr.^{*†}, Daniel Fernandes Macedo[†], Johann M. Marquez-Barja^{*}

^{*}University of Antwerp - imec- IDLab - Antwerp, Belgium

[†]Universidade Federal de Minas Gerais - Computer Science Department - Minas Gerais, Brazil

E-mail: {gilson.miranda, johann.marquez-barja}@uantwerpen.be, damacedo@dcc.ufmg.br

Abstract—An increase of Video on Demand (VoD) consumption has occurred in recent years. Delivering high Quality of Experience (QoE) for users consuming VoD is crucial. Many methods were proposed to estimate QoE based on network metrics, or to obtain direct feedback from video players. Recent proposals usually require monitoring tools installed in multiple network nodes, instrumentation of client devices, updates on existing network elements, among others. We propose a method based on Internet Control Message Protocol (ICMP) probing to estimate QoE for users consuming VoD. The method allows network operators to estimate which QoE level can be delivered to the user according to current network conditions using a Machine Learning (ML) model. Our method does not require installation of software at different network nodes, relying on ICMP probing which is widely supported by existing devices. Our QoE inference model estimates Mean Opinion Score (MOS) with Root Mean Square Error (RMSE) of 0.98, with additional 27 Kbps of traffic during probing. We evaluate the model's generalization capacity when estimating QoE for videos different from the one used for training, which can speed up model's creation process. In those cases MOS was estimated with RMSE of 1.01.

Index Terms—quality of service, quality of experience, DASH video

I. INTRODUCTION

Assessment of user satisfaction with networked services gained prominence recently. Network and Over The Top (OTT) operators have already acknowledged that users' satisfaction with services and connectivity is poorly expressed using traditional Quality of Service (QoS) metrics [1]. Industry and academia have turned to QoE, which indicates the users' degree of satisfaction or annoyance when consuming services or applications [2]. QoE assessment for VoD is a relevant topic, as this kind of application will account for over 80 % of global IP traffic by 2022 [3]. Most internet video traffic is now delivered through HTTP Adaptive Streaming (HAS) [4], [5]. HAS adapts video quality during playback according to network conditions to avoid playback stalling and rebuffering events. Standardization efforts have been carried to develop an open standard for HAS, resulting in the Dynamic Adaptive Streaming over HTTP (DASH).

Many works in the literature seek to estimate QoE for DASH VoD [6]. The first group of methods monitors client applications collecting data such as buffer usage and playback stalls, providing reliable and accurate information. This kind

of method is commonly used for local tuning of player applications, with no transmission of this information to the network, which could raise privacy concerns. A second group monitors network QoS such as throughput and Packet Loss Rate (PLR), relying on information that can be extracted from network flows or inspected from packet data. This type of method can be costly according to the amount of information needed to be extracted. It also raises privacy concerns and can be infeasible for encrypted traffic. A third group of methods combine application and network information, and can provide accurate QoE information but suffers from privacy and limitation problems found in both methods. While some proposals use high-level QoS obtained using specialized tools [7], others require inspection of TCP/HTTP headers [8], [9]. With no consolidated solution to monitor QoE, the heterogeneity poses an additional challenge for network operators.

Feedback-based extensions to DASH have been specified, namely, Server and Network Assisted DASH (SAND) [10]. SAND gives the ability for clients and network elements to exchange signaling messages to improve user QoE. SAND, however, presents many hurdles for widespread adoption: it requires network elements prepared to process the signaling messages and perform eventual adjustments [11]; there is no standard format or metrics to map between network QoS and QoE [1]; existing network elements may be incompatible with SAND.

We propose a method using ICMP probing to monitor network QoS between DASH client and server. Being a well supported protocol, it works out of the box on legacy network equipment. We developed a simple algorithm to adjust probing frequency and run multiple parallel probes to obtain the appropriate granularity of measurements. Therefore, the operator can obtain QoS conditions of Round-Trip Time (RTT), jitter and PLR. Such measurements are passed to a ML model that estimates the delivered QoE in terms of MOS according to the ITU-T P.1203 Recommendation [12]. The dataset used to train the model was created using a controlled environment with a catalog of 19 videos with different contents and duration. Samples from the video with shortest duration were used for model training, while samples from the remaining videos were used to evaluate generalization. The model provided inferences of MOS with RMSE of 1.01.

The paper is organized as follows: Section II discusses related work, focusing on those performing QoE inference for

VoD based on network QoS. Section III describes the proposed method. Setup of experiments is detailed in Section IV. Results are presented and discussed in Section V. Section VI presents the conclusion and future work.

II. RELATED WORK

Costa et al. [7] propose the use of network measurements to estimate Application QoS (AQoS), and from AQoS predict user's QoE. The first step is mapping delay, jitter, throughput and PLR to startup time, stall count, and total stall time. Network is monitored using NetMetric [13], requiring probes in multiple points. The experiments use videos with fixed resolutions (1080p and 720p). The authors in [8] reconstruct a video session analyzing packets passing through an intermediate node. They use this information to determine QoE parameters such as rebuffering events and bit rate variation. The system also requires extraction of the manifest file to process the intercepted information.

The approach in [9] inspects packets of the video flow to find those carrying video segments, also using an HTTP proxy to overcome flow encryption. An algorithm estimates initial playback delay, number and duration of rebuffering events. Experiments were performed using a single video with fixed quality. Khokhar et al. [14] present a method to estimate QoE using network-level measurements on encrypted YouTube traffic. In addition to features such as throughput, packet interarrival times and chunk sizes, another 48 features are used. QoE is estimated in terms of playback status, quality switches and MOS. The work does not address how to obtain all the input features in real networks.

Our method is based on simple ping tools, eliminating the demand for specialized monitoring software. Further, active probing provides better privacy than methods based on packet inspection, and is applicable to encrypted VoD services. Our method provides a more comprehensive indication of user QoE, as it estimates more end-user metrics. It estimates MOS based on ITU-T P.1203 Recommendation [12], comprising metrics such as video stalls, video quality switches and video resolution played. Model training is more streamlined than the state of the art, because it employs data samples from a single video, at the same time reaching similar accuracy levels.

III. PROPOSED METHOD

Our method is composed by an ICMP probing module that estimates QoS conditions between VoD client and server, and an MOS inference model based on decision trees. One way to apply the method is shown in Figure 1. The Probing Module (PM) is co-located with the VoD server, considering that the monitoring takes place at the same network point as the VoD server and takes the same route. This restriction is feasible in the context of partnerships between Internet Service Providers (ISPs) and OTT using CDN-ISP or Mobile Edge Computing (MEC) [15], [16]. The resulting QoS is passed to the QoE Model that estimates the MOS value.

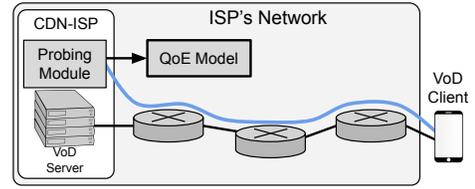


Fig. 1. Probing module deployed along the VoD server (CDN-ISP or MEC context).

A. Probing Module

An initial analysis of the collected data showed that the MOS was strongly affected by PLR in downstream (server to client). In our samples the sessions with 0 % of PLR had an average MOS of 5, while those with 0.5 % PLR had average MOS close to 3. Above 2.3 % PLR the average MOS was approximately 2. Because of the strong influence of PLR we defined that our PM must provide PLR with 0.1 % granularity, requiring 1,000 samples to be able to achieve such granularity. The amount of samples to be collected by the PM can be relaxed to explore the trade-off between precision and traffic overhead. The required samples must be collected within a limited and recent window of time, in order to reflect the current state of the network. This work uses a window of 30 seconds, which is twice the size of typical buffers in DASH clients. The PM spawns multiple parallel and independent ping probes. The probing frequency of all probes is adjusted according to observed RTT and jitter, in order to obtain the required amount of data during the time window.

Traffic overhead generated during probing can be estimated using $O_{dir} = \frac{minSamples}{timeWindow} \times (S_{icmp} + P_{ov})$ where O_{dir} is the overhead per second on a given direction (downstream or upstream). $minSamples$ is the minimum amount of samples to collect during a window of time, $timeWindow$ is the duration of the time window in seconds. S_{icmp} is the data size of probe packets in bits and P_{ov} is the protocol overhead from underlying technologies.

B. MOS Inference Model

We use an ML model to perform MOS inferences from the QoS input data provided by the PM. We selected eXtreme Gradient Boosting (XGBoost) [17], as methods based on Decision Tree (DT) have shown better results (predictions with lower RMSE) when mapping QoS to QoE [18]. XGBoost requires a labeled dataset, which was created instrumenting the DASH client to collect three execution metrics: i) *Representation*, indicating the resolution and bitrate of the video segment being played; ii) *Playback rate*, that returns if the playback is running or if the video is stalled; iii) *Timestamp*, to mark the time each measurement was taken. All values were stored in intervals of 0.5 seconds. These values are used to calculate the MOS values based on the ITU-T P.1203 Recommendation.

IV. EXPERIMENT SETUP

The infrastructure used to generate the dataset is shown in Figure 2. We set up three Docker containers. The server runs

the NGINX Server [19] hosting 19 different videos and the DASH player application. The client container accesses the server and runs the player over Firefox. We used the reference player provided by DASH Industry Forum¹ version 3.0.0, instrumented to collect playback metrics. The configurations of the player were kept as default, except for buffer which was changed to 12 seconds so network oscillations would be reflected quicker in playback quality. A custom script in the client stored the metrics locally to avoid generating additional network traffic. The last container is the QoS Monitor, which performs ping probing based on the fping tool². Network impairments were set using Traffic Control (TC). Server and QoS Monitor are considered as deployed at the same network point, therefore the same impairment values were set to their interfaces.

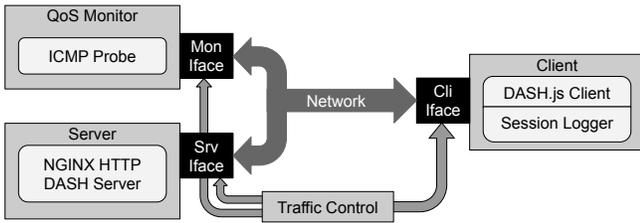


Fig. 2. Experimental environment

TABLE I
SAMPLE VIDEOS

Video	Duration	Type
¹ Another World (another)	00:03:11	Nature
¹ Another World 2 (another2)	00:03:06	Nature
¹ Football Barcelona (barcelona)	00:03:14	Sports
¹ The Fountains Of Bellagio (bellagio)	00:03:43	Arts
¹ La Boheme (boheme)	00:04:29	Music Video
¹ Power of Curve (curve)	00:03:15	Promotional
¹ The Quiet Czech (czech)	00:03:24	Documentary
¹ Phantom Flex (flex)	00:03:07	Promotional
¹ Garden (garden)	00:03:05	Promotional
¹ Jimix Put Your Hands Up (jimix)	00:03:56	Music Video
¹ Landscape (landscape)	00:03:10	Nature
¹ Lumix (lumix)	00:03:07	Documentary
¹ Slam Dunk (slam)	00:02:56	Sports
¹ Surfing (surfing)	00:02:59	Sports
¹ Lovely Swiss (swiss)	00:03:41	Documentary
¹ Travel With My Pet (travel)	00:02:35	Documentary
¹ See the Unexpected (unexpected)	00:03:18	Sports
¹ Life Untouched (untouched)	00:03:18	Nature
¹ 7 Wonders Of The World (wonders)	00:03:51	Documentary

¹<http://4kmedia.org>

The sample videos are described on Table I. A short name in parenthesis, used as reference throughout this paper, is shown for each video. The table also shows the duration and type of each video. All videos were prepared following a standard process. Each original video was encoded in 10

representations (a version of the video with a specific resolution and bitrate), then, each version is split into segments of short duration (4 seconds in our work). Finally, a Media Presentation Description (MPD) file is generated containing all information needed for the client application to download and properly adapt playback quality according to observed network conditions. Our videos were encoded using the H.264 codec, without an audio track. The 10 representations used were: 320x180 (200 Kbps), 320x180 (400), 480x270 (600), 640x360 (800), 640x360 (1,000), 768x432 (1,500), 1024x576 (2,500), 1280x720 (4,000), 1920x1080 (8,000), 3840x2160 (12,000). The MPD bandwidth field was fixed for each representation in all videos, in order to the client operate consistently. The values (in bits per second) used were, respectively: 256K, 512K, 760K, 1,020K, 1,260K, 1,900K, 3M, 4M, 10M, 20M.

Bandwidth and delays set in TC were taken from a uniform distribution between 0 and 400 Mbps, and 0 and 800 ms, respectively. Jitter was uniformly taken between 0 and half of the delay used in the session. PLR values were set according to a Gamma distribution with shape $k = 0.3$, and scale $\theta = 1$, derived to achieve a similar distribution to measurements provided by *Measurement Lab (M-Lab)*³. For MOS calculation and dataset labeling according to ITU-T P.1203 Recommendation we used the software⁴ provided by [20], [21]. The operation mode used was 0, combining video resolution (and resolution switches), and occurrence and duration of playback stalls. Other parameters for the software were **device type** (PC), **display resolution** (3840x2160), and **viewing distance** (150 cm).

V. RESULTS

A. Data Analysis

We performed over 60,000 video sessions. Approximately 25,000 sessions were executed with the “travel” video due to its shorter duration, which allowed us to experiment a wider range of network impairments. Approximately 2,000 sessions were executed for each other video. Table II shows the Spearman correlation between QoS and MOS. The first eight rows show the correlations with parameters set using TC and the last three with measured QoS. Although the table shows a weak correlation between Downlink Bandwidth and MOS, it occurs due to range of values used for experiments. For sessions with bandwidth up to 4 Mbps, the observed correlation was 0.61. However, correlations between Downlink Bandwidth and MOS quickly drop as we evaluate sessions with higher bandwidth values (e.g. 0.15 for sessions with up to 40Mbps). It should be noted that TC bandwidth configuration is not directly reflected in throughput, therefore, even sessions configured for 400 Mbps can experience low throughput due to delay, jitter and PLR. Different from bandwidth, Downlink PLR and MOS show a strong correlation for the entire evaluated range. Correlations with the measured metrics also show a strong influence of PLR on MOS, followed by RTT.

¹<https://dashif.org/>

²<https://fping.org/>

³<https://www.measurementlab.net/>

⁴<https://github.com/itu-p1203/itu-p1203>

TABLE II
QoS AND MOS SPEARMAN CORRELATION

QoS Condition	Correlation to MOS	p-value
Downlink Bandwidth	0.0226	$p < 0.001$
Uplink Bandwidth	0.0007	0.0182
Downlink Delay	-0.1872	$p < 0.001$
Uplink Delay	-0.1354	$p < 0.001$
Downlink Jitter	-0.1825	$p < 0.001$
Uplink Jitter	-0.1128	$p < 0.001$
Downlink PLR	-0.7752	$p < 0.001$
Uplink PLR	0.0093	$p < 0.001$
RTT	-0.2021	$p < 0.001$
Jitter	-0.1456	$p < 0.001$
PLR	-0.4567	$p < 0.001$

B. Model Training

For model training we used samples of the video with shortest duration (“travel”), from which we had more sessions with different network conditions. We used data of 20,000 sessions of “travel” for hyperparameter tuning, training and Cross Validation (CV). Approximately 5,000 sessions of “travel” and all sessions of the other videos were used for evaluation. Hyperparameter tuning was done using 100 trials of random search [22]. The hyperparameter values selected for XGBoost were: *colsample_bytree* of 0.85, *colsample_bylevel* of 0.8, *subsample* of 0.91, *learning rate* of 0.02, *alpha* of 1, and *max_depth* of 20. The maximum number of trees was set to 1,000, and we used early stopping to interrupt training after 20 successive rounds of no accuracy improvement.

C. Inference Results

The RMSE obtained in 5-fold CV was **0.8965** with standard deviation of **0.0003**. Over the generalization dataset, the RMSE for the “travel” video using the final model was **0.9887**. For sessions of videos different than “travel”, the overall RMSE was **1.0131**. Table III shows the RMSE obtained for each individual video. These results show slightly different RMSE values for each video, with an average of **1.0052** and standard deviation of **0.0231**. Minor differences are possibly caused by video content differences, which make the file size of video segments to vary between videos. Nevertheless, RMSE values were similar to those obtained with “travel”, with an RMSE oscillation between -0.02 and $+0.06$. With such small differences, an operator can use a single model to infer MOS for all videos on the server. The creation of the model is accelerated as the short-duration video allows more sessions with distinct network impairments to be executed. Also, retraining is needed only when a new video format (e.g. different resolutions or codecs) is added.

Figure 3 shows the error according to MOS range. Higher errors occurred when the MOS was high (between 4 and 5). On the other hand, when MOS was below 4 the distribution of RMSE values is similar for all classes of MOS, with errors below 1 in approximately 80 % of samples. This shows a pessimistic behavior of the system, inferring a low QoE when the client is actually receiving a high QoE. On the other hand,

TABLE III
INFERENCE RMSE BY VIDEO

Video	RMSE	Video	RMSE	Video	RMSE
another	0.98	another2	0.98	barcelona	1.02
bellagio	1.01	boheme	1.04	curve	1.02
czech	1.01	flex	0.97	garden	1.01
jimix	1.03	landscape	1.03	lumix	1.00
slam	0.96	surfing	0.99	swiss	1.03
travel	0.98	unexpected	1.00	untouched	1.01
wonders	1.03	-	-	-	-

higher accuracy is achieved for lower QoE levels. This result is a consequence of the PM’s inability to differentiate downlink from uplink packet loss.

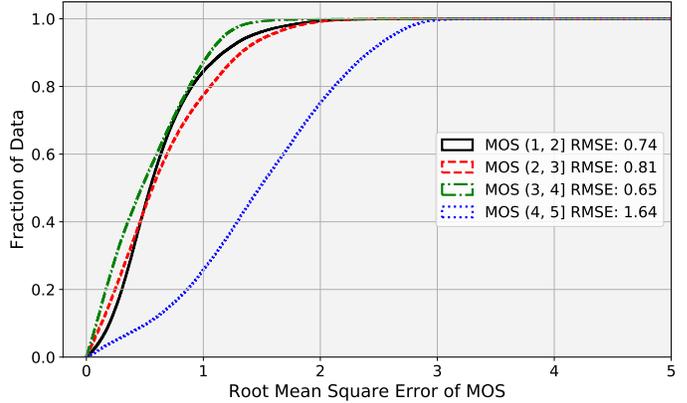


Fig. 3. MOS inference error by MOS class

VI. CONCLUSION AND FUTURE WORK

We proposed a practical method for QoE inference for DASH VoD that does not require instrumentation of client devices, changes on existing network elements, deep flow inspection or proprietary tools. Monitoring is done using widely supported ICMP. An ML model was trained to infer MOS based on the ITU-T P.1203. For model training we used data from sessions of the shortest video of the catalog, and evaluated model’s accuracy for the other ones, achieving an RMSE of 1.01. Probing overhead was minimal, taking 1.4 % of traffic if the video were served in lowest quality. In future work we will use this method as feedback for network control.

ACKNOWLEDGMENT

This study was partially financed by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, CNPq (funding agency from the Brazilian federal government), and FAPEMIG (Minas Gerais State Funding Agency). This work has partially received funding from the European Union’s Horizon 2020 Fed4FIREplus project under grant agreement no.723638.

REFERENCES

- [1] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of quality of experience of video-on-demand services: A survey," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 401–418, 2016. [Online]. Available: <https://doi.org/10.1109/COMST.2015.2401424>
- [2] I. Rec, "P. 10: Vocabulary for performance and quality of service, amendment 2: New definitions for inclusion in recommendation itu-t p. 10/g. 100," *Int. Telecomm. Union, Geneva*, 2008. [Online]. Available: <https://www.itu.int/rec/T-REC-P.10>
- [3] Cisco, "White paper: Cisco Visual Networking Index: Forecast and Trends, 2017–2022," *Cisco*, 2018. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1213-business-services-ckn.pdf
- [4] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hofbeld, 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. [Online]. Available: <https://doi.org/10.1109/COMST.2014.2360940>
- [5] Sandvine, "Global Internet Phenomena Report: Latin America and North America," 2016, Accessed: Jul. 1, 2020. [Online]. Available: <https://www.sandvine.com/hubfs/downloads/archive/2016-global-internet-phenomena-report-latin-america-and-north-america.pdf>
- [6] N. Barman and M. G. Martini, "QoE Modeling for HTTP Adaptive Video Streaming-A Survey and Open Challenges," *IEEE Access*, vol. 7, pp. 30 831–30 859, 2019. [Online]. Available: <https://doi.org/10.1109/ACCESS.2019.2901778>
- [7] R. I. T. Da Costa Filho, W. Lautenschlager, N. Kagami, V. Roesler, and L. P. Gaspary, "Network fortune cookie: Using network measurements to predict video streaming performance and QoE," *2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings*, pp. 2–7, 2016. [Online]. Available: <https://doi.org/10.1109/GLOCOM.2016.7842022>
- [8] R. Huysegems, B. De Vleeschauwer, K. De Schepper, C. Hawinkel, T. Wu, K. Laevens, and W. Van Leekwijck, "Session reconstruction for HTTP adaptive streaming: Laying the foundation for network-based QoE monitoring," in *2012 IEEE 20th International Workshop on Quality of Service*, 2012, pp. 1–9. [Online]. Available: <https://doi.org/10.1109/IWQoS.2012.6245987>
- [9] C. Ge and N. Wang, "Real-time QoE estimation of DASH-based mobile video applications through edge computing," *INFOCOM 2018 - IEEE Conference on Computer Communications Workshops*, pp. 766–771, 2018. [Online]. Available: <https://doi.org/10.1109/INFOCOMW.2018.8406935>
- [10] E. Thomas, M. O. Van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey, "Enhancing MPEG DASH Performance via Server and Network Assistance," *SMPTe Motion Imaging Journal*, vol. 126, no. 1, pp. 22–27, 2017. [Online]. Available: <https://doi.org/10.5594/JMI.2016.2632338>
- [11] A. A. Barakabitze, N. Barman, A. Ahmad, S. Zadtootaghaj, L. Sun, M. G. Martini, and L. Atzori, "QoE management of multimedia streaming services in future networks: A tutorial and survey," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, pp. 526–565, 2020. [Online]. Available: <https://doi.org/10.1109/COMST.2019.2958784>
- [12] ITU-T Telecommunication Standardization Sector, "ITU-T Rec P.1203: Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport," 2017. [Online]. Available: <https://www.itu.int/rec/T-REC-P.1203>
- [13] G. L. Dos Santos, V. T. Guimarães, J. G. Silveira, A. T. Vieira, J. A. De Oliveira Neto, R. I. T. Da Costa Filho, and R. Balbinot, "UAMA: A unified architecture for active measurements in IP networks End-to-end objective quality indicators," *10th IFIP/IEEE International Symposium on Integrated Network Management 2007, IM '07*, pp. 246–253, 2007. [Online]. Available: <https://doi.org/10.1109/INM.2007.374789>
- [14] M. J. Khokhar, T. Ehlinger, and C. Barakat, "From network traffic measurements to QoE for internet video," *IFIP Networking Conference*, pp. 1–9, 2019. [Online]. Available: <https://doi.org/10.23919/IFIPNetworking.2019.8816854>
- [15] B. Frank, I. Poese, Y. Lin, G. Smaragdakis, A. Feldmann, B. Maggs, J. Rake, S. Uhlig, and R. Weber, "Pushing CDN-ISP collaboration to the limit," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 34–44, 2013. [Online]. Available: <https://doi.org/10.1145/2500098.2500103>
- [16] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017. [Online]. Available: <https://doi.org/10.1109/MCOM.2017.1600863>
- [17] 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: <https://doi.org/10.1145/2939672.2939785>
- [18] S. Aroussi and A. Mellouk, "Statistical evaluation for quality of experience prediction based on quality of service parameters," *International Conference on Telecommunications (ICT)*, pp. 1–5, 2016. [Online]. Available: <https://doi.org/10.1109/ICT.2016.7500364>
- [19] W. Reese, "Nginx: The high-performance web server and reverse proxy," *Linux J.*, vol. 2008, no. 173, Sep. 2008. [Online]. Available: <https://doi.org/10.5555/1412202.1412204>
- [20] 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: <https://doi.org/10.1109/QoMEX.2017.7965631>
- [21] 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: <https://doi.org/10.1145/3204949.3208124>
- [22] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *J. Mach. Learn. Res.*, vol. 13, no. null, p. 281–305, Feb. 2012. [Online]. Available: <https://doi.org/10.5555/2188385.2188395>