QoEyes: Towards Virtual Reality Streaming QoE Estimation Entirely in the Data Plane

Francisco Vogt*, Fabricio Rodríguez Cesen*, Ariel Góes de Castro+
Marcelo Caggiani Luizelli+, Christian Esteve Rothenberg*, Gergely Pongrácz|
+Federal University of Pampa (UNIPAMPA), Brazil
*University of Campinas (UNICAMP), Brazil
|Ericsson Research, Hungary

2023 IEEE 9th International Conference on Network Softwarization (NetSoft) | 979-8-3503-9980-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/NetSoft57336.2023.10175463

***Abstract*—In recent years, advances in virtual reality (VR) technologies (e.g., high-quality VR headsets) have enabled a new perspective of experiences for users (e.g., gaming, online events). However, ensuring the user experience is still a challenge. Existing solutions are limited to measuring and estimating QoE at the user plane (e.g., VR player) or at the control plane, imposing unfeasible latency for different scenarios (5G networks and beyond). In this work, we propose QoEyes, an in-network QoE estimation based on the use of Inter-Packet-Gap (IPG) in programmable devices. Our results show that the IPG measured on the data plane is strongly linked to QoE, yielding an accurate data plane QoE estimate.**

I. Introduction

Virtual Reality (VR) video streaming applications are already a reality. VR head-mounted displays are expected to grow to nearly 34 million by the end of 2023, while the associated network traffic is expected to increase at least 12-fold [1]. It poses a significant challenge to network operators as VR video streaming applications will demand stringent network performance to achieve a reasonable Quality of Experience (QoE). Recent studies [2] indicate that VR video applications require network delay lower than 9ms, while the bandwidth requirements can surpass 500 Mbps.

To reduce bandwidth requirements, VR video streaming players rely on spherical-to-plane projection such as the tile-based scheme [3]. In this approach, VR videos are encoded at different resolutions (e.g., 720p, 1080p, 4K) and then split into spatial (a.k.a tiles) and temporal segments. During the streaming, the VR player only requests segments and tiles corresponding to the visible area of the full 360-degree panoramic view (a.k.a. viewport). Additionally, the VR player relies on other strategies such as Adaptive Bitrate and Buffer Management heuristics, which allows requesting segments that are estimated to belong to the viewport in higher quality, while the other tiles are requested at lower resolutions.

This work was supported by the Innovation Center, Ericsson S.A. and by the Sao Paulo Research Foundation (FAPESP), grant 2021/00199-8, CPE SMARTNESS. This study was partially funded by CAPES, Brazil - Finance Code 001. This work was partially funded by National Council for Scientific and Technological Development (CNPq 404027/2021-0), Foundation for Research of the State of Sao Paulo (FAPESP 2021/06981-0, 2021/00199-8, 2020/05183-0), and Foundation for Research of the State of Rio Grande do Sul (21/2551-0000688-9).

Recent studies in this domain have focused on measuring and estimating the QoE of VR video streaming. However, they are typically done at the user plane (e.g., directly at the VR player) or the control plane. For instance, [2] propose a user plane two-stage ML-assisted approach that infers how the users perceive the streamed VR video performance. Despite a few initiatives, little has been done to directly infer the QoE in the data plane. It would bring the benefit of analyzing each VR video session QoE and reacting to different network conditions in near real-time.

In this paper, we introduce QoEyes, an in-network QoE estimation. QoEyes relies on the use of Inter-Packet-Gap (IPG) in a programmable network device. The IPG refers to the arrival time difference between two consecutive network packets of the same network flow. By measuring the IPG per each VR video session, the network can infer QoE directly on the data plane and thus make decisions in order to improve it. Figure 1 illustrates an overview of QoEyes, which performs QoE estimation on programmable network devices (by analyzing the IPG of video sessions) and relays this information to the control plane as required. We implemented QoEyes in P4 using Barefoot Tofino hardware and evaluated using publicly available VR video traces. Results show that QoEyes can strongly correlate calculated IPG with the achieved Mean Opinion Score (MOS). The main contributions of this paper can be summarized as follows:

- the design of an in-network VR video streaming QoE estimation directly in the data plane;
- a prototype implementation using Barefoot Tofino hardware;
- open-source software artifact for reproducibility[1];

The remainder of this paper is organized as follows. Section 2 discusses the background and related work in VR video streaming. Section 3 introduces the QoEyes design in programmable data planes. In Section 4, we present and discuss the results of an evaluation of the proposed approach. Last, in Section 6, we conclude the paper with final remarks and perspectives for future work.

[1]https://github.com/intrig-unicamp/QoEyes

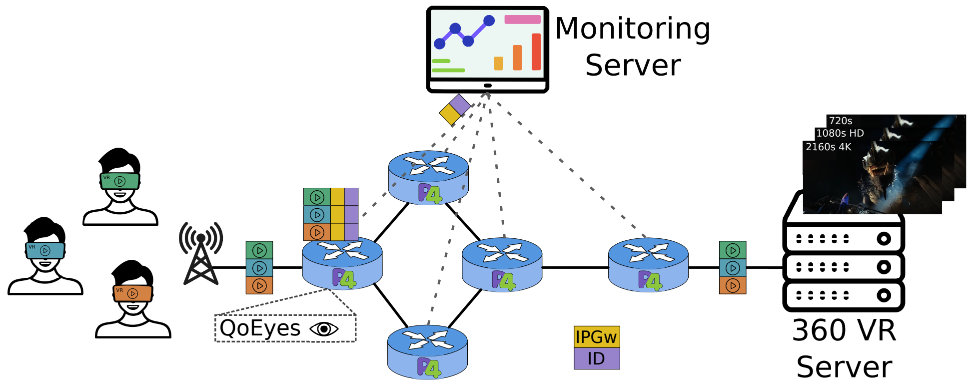


Fig. 1. Illustrating an use case of `QoEyes` in a P4-based network.

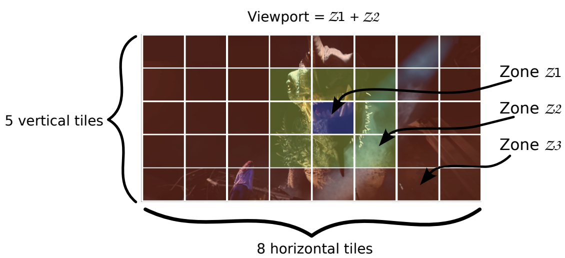


Fig. 2. Three zones 8×5 tiling scheme.

II. Background & Related Work

A. QoE and VR Video Streaming

QoE measurement is subdivided into subjective and objective approaches. The first measures the video quality perceived by the Human Visual System (HVS) and is more accurate than the objective assessment methods [4]. However, these methods are manually set (i.e., offline) based on user observation and feedback through their behavior (e.g., presence, motion sickness, perceptual quality). On the other hand, objective metrics are based on quantifiable data that can be measured or calculated (e.g., bandwidth and latency). To better capture the user's experience, it is often best to use subjective and objective metrics. For example, we can collect objective metrics from inferring user experience by mapping objective metrics to the subjective experience. This mapping can be established through user studies or statistical models developed based on user studies' data.

A real scenario where QoE methods can be applied is VR video streaming. VR video streaming refers to delivering 360-degree video content that users can interact with and explore as if they were in the environment. Unlike traditional 2D video, it provides a fully immersive experience that allows users to look around and explore the virtual world from different perspectives. However, VR video streaming faces several technical challenges, such as high bandwidth requirements, low image quality, and limited interactivity. These difficulties stem from the need to project/decompose the video into tiles for the user. Figure 2 summarizes an 8x5 tiling scheme. In summary, we break down the VR video into smaller tiles or fragments, encoding each tile separately, and then, the tiles are transmitted over the network. The goal of the tiling is to provide the viewer with the ability to view a high-resolution VR video without having to download the entire video beforehand. To that end, the tiling scheme also allows for the use of adaptive bitrate (ABR) algorithms, which adjust the bitrate of each tile based on the network conditions to ensure smooth playback. For example, Zone $z1$ is defined as containing only the viewport's central tile, Zone $z2$ encompasses the viewport border tiles (8 tiles), and Zone $z3$ has the 31 remaining tiles. To overcome these challenges, developers are improving VR video compression algorithms, hardware, and software solutions. To reduce the high bandwidth demands, the user viewport is limited to the portion of the virtual environment that the user can see at a given time and is defined by the field of view (FOV) of the VR headset or device, which determines the size and shape of the virtual environment that is visible to the user. With that in mind, a key aspect in the future of VR is selecting a set of objective metrics (e.g., Quality of Service (QoS) metrics). However, little has been done [5], [6] towards effective metrics to guarantee a better QoE in real-time.

B. Inter-Packet Gap (IPG)

The IPG is a promising network metric that has proven recently to be efficient in solving several network problems such as microbursts [7] and heavy hitters detection [8]. Traditionally, the IPG refers to the arrival time difference between two consecutive network packets. and can be calculated according to the Equation 1, where TS_l and TS_p are the arrival time of the last and penultimate packet, respectively.

$$IPG = TS_l - TS_p \tag{1}$$

Despite being a simple metric to be calculated when combined with other mathematical methods such as Cumulative Sum (CUSUM), Exponentially Weighted Moving Average (EWMA), and Double Exponentially Weighted Moving Average (DEMA), the IPG has already proven to be a powerful metric for QoE inference [9]. In this work, we argue that by calculating the IPG using an EWMA in a programmable network device, we can get a QoE estimation directly on the data plane and thus make decisions in real-time, improving the user's QoE. More details are discussed in the next chapter.

C. Related Work

VR-EXP [3] is a platform that enables a set of adaptive tile-based schemes for various network conditions. PREDICTIVE [2] is a two-stage ML-assisted approach that infers how the user perceives the resulting VR video playout performance. In summary, a set of predictors have the network QoS (i.e., delay, packet loss, and TCP throughput) and the tiling scheme as input. In contrast, Vidhya et al. [10] introduced a fuzzy logic mechanism within the Network Data and Application Function (NWDAF) entity in 5G to perform QoE evaluation. Similarly, Schwarzmann et al. [11] leverages NWDAF standardized interface capabilities in 5G networks to estimate the accuracy of different state-of-the-art regression techniques. Chen et al. [12] propose an SSIM-based approach for 360-degree video quality assessment. The algorithm explores a correlation between 2D and spherical projection. Also, it is verified on a subjective 360-degree video quality assessment database. On the other hand, FastInter360 [13] exploits a set of texture features to reduce the encoding time of 360-degree videos with equirectangular projection (ERP) while Upenik et al. [14] extends Yu et al. [5] and benchmarks PSNR-based approaches against ground-truth subjective quality data.

Iurian et al. [15] analyzed the impact of priority queues in a video streaming scenario. However, the work is preliminary and does not provide QoE inference. Conversely, Bhat et al. [16] leverage Q-in-Q tunneling and translates the application- into link-layer header information at the edge to infer objective QoE metrics and the model QoE value score of VR streaming sessions.

Despite existing research efforts to achieve QoE directly in the data plane, to the best of our knowledge, `QoEye` is the first approach toward designing an in-network VR video streaming QoE estimation directly in the data plane.

III. Inferring QoE in the Data Plane

Typically, the QoE is estimated by utilizing methods of measurement or prediction in either the user plane or the control plane side. These methods are known for their accuracy in estimating the user's QoE and generally involve measuring various QoS metrics and utilizing machine learning or deep learning algorithms. Despite providing accurate results, these techniques may not always be as quick in their measurement due to the extent and complexity of the network metrics that need to be measured and the efficiency of the machine learning or deep learning algorithms employed. As a result, it becomes difficult to promptly implement network policies to address potential problems or degradation of quality.

To address this limitation, in this work, we propose a first step to the QoE estimation entirely in the data plane. The data plane QoE assessment allows making decisions at a nanosecond level, enhancing the reaction time to problems and thereby improving the user QoE. For instance, if QoE loss is detected, strategies like [15] can be user for flow prioritization and QoE improvement.

The principal challenges of performing QoE inference on the data plane are: (i) hardware resource and operation

Control `IngressPipeline():`
- **RegisterAction** `ComputeIPG(`IPG_c`):`
  - IPG_w = α * IPG_w + (1 - a) * IPG_c
  - **return** IPGw
- **Register**<**bit**<16>> IPG_w
- **Register**<**bit**<32>> TS_l
- **Apply:**
  - IPG_c = TS_c - TS_l
  - IPG_w = ComputeIPG(IPG_c)
  - PacketClone()

Control `EgressPipeline():`
- **if** *Cloned Packet* **then**
  - InsertIPGHeader()
- **end**

Algorithm 1: QoEyes data plane algorithm

constraints; and (ii) limited information about data flow. In the P4 programmable hardware targets, such as the Tofino Native Architecture (TNA), we have limited storage capacity, memory access restrictions, and limitations with arithmetic and comparison operations. For example, the register can only be accessed once during the lifespan of a packet, comparisons are restricted to a set number of bits, and arithmetic operations such as division and multiplication can only be performed using bit shifting. Additionally, the data plane has limited knowledge about the data flow. It does not possess information about the application level, including aspects such as video resolution, buffer size, and segment size.

Therefore, we introduce `QoEyes`, a QoE estimation technique that uses IPG calculation to carry out the QoE estimation entirely in the data plane. Recent academic studies have already utilized IPG as the primary metric for inferring QoE, demonstrating a direct correlation between IPG and QoE [9]. We argue that we can get a QoE estimation directly in the data plane by calculating the IPG as an EWMA.

Consider the set of network flows F containing all streams (or VR video sessions) from the network. Then, for each packet $p \in F$, we calculate the IPGw (Inter Packet Gap weighted) using the EWMA described in Equation 2. In the equation, IPG_c represents the IPG measured with the Equation 1, IPG_{w-1} the last calculated IPGw, and α is an input parameter to control the IPGw variation.

$$IPG_w = \alpha \cdot IPG_{w-1} + (1 - \alpha) \cdot IPG_c \quad (2)$$

The Algorithm 1 illustrates how we implemented the `QoEyes` strategy to perform the IPGw measurement in the Tofino architecture. We perform all the calculations in the ingress pipeline, using registers to store the IPGw and the last noted timestamp (TS_l). In addition, we use register actions to perform the approximate EWMA calculation, which is done through bit operations. Also, we performed a clone of the packet in the ingress pipeline and inserted the IPG header into the cloned packet. This IPG header has the measured IPGw and the corresponding flow ID and is reported to the control plane. In practical applications, these reports can be conducted at regular intervals or only when necessary to monitor the

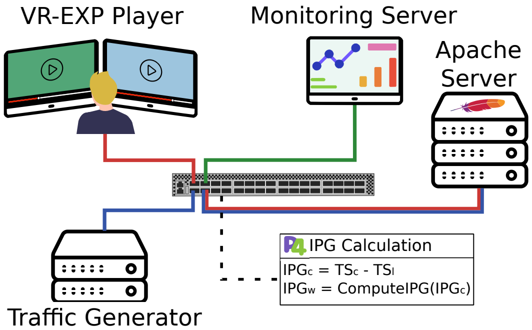


Fig. 3. Experimental environment.

IPGw. Additionally, the reporting can be triggered only when a drift in the IPGw is detected, which signifies a QoE decline.

IV. Experimental Evaluation

A. Setup

Testbed. We evaluate the performance of `QoEyes` using a *Barefoot Tofino Switch* (Edgecore Wedge 100BF-32X) and four servers (Intel Xeon E5-2620v2, dual-port 10G Intel X540-AT2 NIC, and 64GB of memory running Ubuntu 20.04) connected via 10G SFP+ interfaces. Figure 3 illustrates the setup of our testbed. For the experiments, one server runs the VR-EXP player [3] to request 360-degree video content, while another is equipped with an Apache server that hosts the VR video content. The traffic between these two servers (i.e., VR clients and server) is forwarded through a programmable Tofino switch, which calculates and monitors the IPG for each VR session. The switch also reports computed IPGs to the monitoring server, which analyses the results. Finally, we have a traffic generator server to generate multiple video sessions in parallel, as background traffic.

Implementation. `QoEyes` algorithm is implemented using P4 language for the Tofino Native Architecture (TNA). In the code, to calculate and store the IPGw, we use registers and actions with 16-bit of IPG_f^w and 32-bit of TS_f^l for each flow entry (VR video sessions). We assume a capacity for $2k$ register entries and use the value of α = 0.96 (approximately) for the EWMA calculation. The resource utilization of our algorithm on Tofino can be seen in Table 1. In the table, we show the resource utilization of the switch.p4 (baseline of p4 code for switching) and the cost of the extra logic `QoEyes` imposes on the switch.p4 (i.e., the cost of `QoEyes` + switch.p4). Observe that `QoEyes` adds not much to the switch's overall physical resource utilization, with a maximum increase of 4.2% in VLIW actions.

Dataset. For the experiments, we rely on the same traces used by Filho et al. [3]. The publicly available traces comprise two 360-degree videos: Google Spotlight and Freestyle Skiing [17]. Each VR video is encoded with 720p, 1080p, and 4K qualities. VR video sessions are initiated using a random user file (also from [3]), which contains a series of movements (described in axis x, y, z) of user interactions with the VR.

TABLE I
Hardware resource utilization

Resource	Switch.p4	QoEyes
Hash Bits	32.3%	34.2%
SRAM	29.8%	30.6%
TCAM	28.4%	28.4%
VLIW Actions	34.6%	38.8%
Stateful ALUs	15.6%	15.6%

TABLE II
Number of tiles received with 100 sessions in parallel

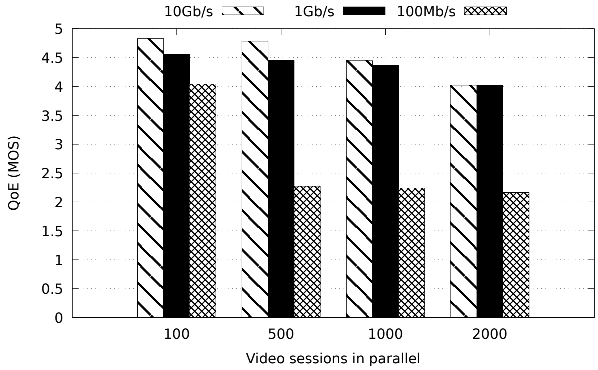
	10Gb			1Gb			100Mb		
	z1	z2	z3	z1	z2	z3	z1	z2	z3
720p	1	12	1375	1	12	1375	1	12	1375
4K	59	468	5	59	468	5	59	468	5

B. Results

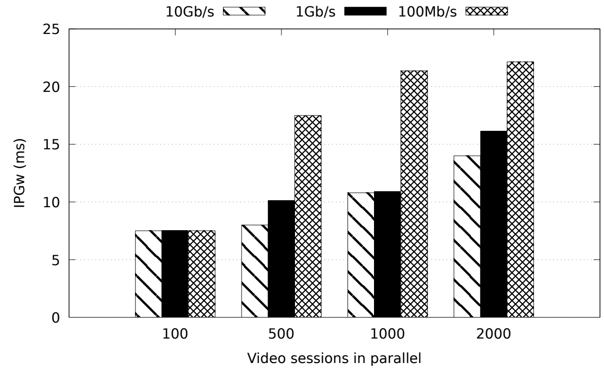
First, we evaluate the relationship between the IPGw measured in the data plane by `QoEyes` and the QoE measured by the user side. For the QoE user side model, we use as a baseline the QoE model provided by Filho et al. [3], which utilizes the output provided by the VR-EXP player to calculate the MOS. In our case, we consider for the calculation of the ideal MOS (MOS = 5), the case where we have only one active video session and with the maximum available link capacity. In all our experiments, we varied the physical port capacity (with Tofino port shaping) of 10Gbps, 1Gbps, and 100Mbps. Also, we varied the number of active VR video sessions running on the network from 100 to 2000. These parameters were applied in our testbed to have different network conditions.

Figure 4 illustrates the IPGw and QoE measured in the experiment by `QoEyes`. As observed in Figure 4(a), the QoE appears to decline with an increase in the number of concurrent active sessions or a decrease in transmission capacity. In Figure 4(b), the IPG shows a similar behavior (but increasing) when the number of active sessions increases or the transmission capacity decreases. The only exception can be seen when there are 100 sessions, where the IPG remains constant, but the QoE varies. It occurs because with a small number of sessions, the traffic can be forwarded in the minimum IPG time (less than 1 ms), and there is only a gap between the establishment and closing of new TCP connections. However, the QoE model employed calculates the QoE based on the bitrate and thus produces better results with higher transmission capacity, even if all tiles arrive with the same quality (see Table II).

In Figure 5, we can see the IPGw behavior over 2000 reports for a flow with a measured QoE MOS greater than 4.5 (blue color) and another flow with a QoE MOS less than 2.5 (red color). Note that most of the time, the flow with QoE below 2.5 has a higher IPG than the flow with good QoE. Furthermore, the IPGw of the stream with lower QoE has higher peaks and a longer recovery time (time to return to a low IPG). In both experiments (Figures 4 and 5), we can observe a strong correlation between the IPGw measured in the data plane, and the QoE MOS measured on the user side.



(a) QoE.



(b) IPGw.

Fig. 4. Comparison between QoE (MOS) and IPGw for an increasing number of VR video sessions.

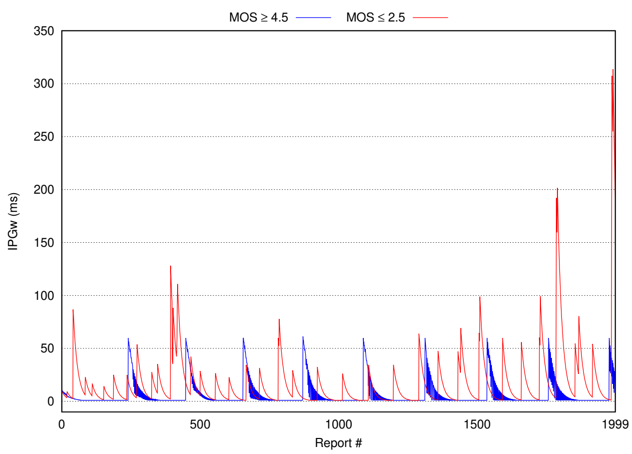


Fig. 5. Calculated IPGw over VR video sessions with different MOS.

V. Final Remarks

In this work, we present `QoEyes`, a full data plane QoE estimation method for 360-degree (VR) videos. `QoEyes` is based on the IPGw calculation and can provide the QoE estimate in near real-time. Our results show a robust correlation between the `QoEyes` measurement in the data plane and the QoE assessed using the MOS function [3]. For our future works, we plan to increase the scope of our evaluation by exploring a wider range of network scenarios and conducting a larger number of experiments. Additionally, we aim to perform experiments on a network-wide scale to determine the QoE across multiple network devices. Finally, we hope to evaluate the feasibility of extending our method to consider additional factors, such as packet size.

References

[1] Cisco, "Cisco visual networking index: Forecast and trends, 2017-2022," Cisco Systems, Tech. Rep., 2019.

[2] R. I. T. da Costa Filho *et al.*, "Predicting the performance of virtual reality video streaming in mobile networks," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 270–283.

[3] R. I. T. D. C. Filho, M. C. Luizelli, S. Petrangeli, M. T. Vega, J. V. d. Hooft, T. Wauters, F. D. Turck, and L. P. Gaspary, "Dissecting the performance of vr video streaming through the vr-exp experimentation platform," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 4, pp. 1–23, 2019.

[4] J. van der Hooft, M. T. Vega, C. Timmerer, A. C. Begen, F. De Turck, and R. Schatz, "Objective and subjective qoe evaluation for adaptive point cloud streaming," in *2020 twelfth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[5] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *2015 IEEE International Symposium on Mixed and Augmented Reality*, 2015, pp. 31–36.

[6] V. Zakharchenko, K. P. Choi, and J. H. Park, "Quality metric for spherical panoramic video," in *Optics and Photonics for Information Processing X*, vol. 9970. SPIE, 2016, pp. 57–65.

[7] F. G. Vogt, F. Rodriguez, C. Rothenberg, and G. Pongrácz, "Innovative network monitoring techniques through in-band inter packet gap telemetry (ipgnet)," in *Proceedings of the 5th International Workshop on P4 in Europe*, 2022, pp. 53–56.

[8] S. K. Singh, C. E. Rothenberg, M. C. Luizelli, G. Antichi, P. H. Gomes, and G. Pongrácz, "Hh-ipg: Leveraging inter-packet gap metrics in p4 hardware for heavy hitter detection," *IEEE Transactions on Network and Service Management*, 2022.

[9] R. U. Mustafa and C. E. Rothenberg, "Machine learning assisted real-time dash video qoe estimation technique for encrypted traffic," in *Proceedings of the 1st Mile-High Video Conference*, 2022, pp. 123–123.

[10] R. Vidhya, P. Karthik, and S. Jamadagni, "Anticipatory qoe mechanisms for 5g data analytics," in *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2020, pp. 523–526.

[11] S. Schwarzmann, C. C. Marquezan, R. Trivisonno, S. Nakajima, V. Barriac, and T. Zinner, "Ml-based qoe estimation in 5g networks using different regression techniques," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3516–3532, 2022.

[12] S. Chen, Y. Zhang, Y. Li, Z. Chen, and Z. Wang, "Spherical structural similarity index for objective omnidirectional video quality assessment," in *2018 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2018, pp. 1–6.

[13] I. Storch, L. Agostini, B. Zatt, S. Bampi, and D. Palomino, "Fastinter360: A fast inter mode decision for hevc 360 video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 3235–3249, 2021.

[14] E. Upenik, M. Rerabek, and T. Ebrahimi, "On the performance of objective metrics for omnidirectional visual content," in *2017 ninth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2017, pp. 1–6.

[15] C.-M. Iurian, R. Botez, I.-A. Ivanciu, and V. Dobrota, "Video streaming evaluation using priority queuing in p4 programmable networks," in *2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2022, pp. 1–5.

[16] D. Bhat *et al.*, "Application-based qoe support with p4 and openflow," in *IEEE INFOCOM*. IEEE, 2019, pp. 817–823.

[17] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in vr spherical video streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 193–198.