

Adaptive 360° Video Streaming Over Wireless Communication Channels

Ioanna Valiandi
VIDEOMICS Group
CYENS Centre of Excellence
 Nicosia, Cyprus
 i.valiandi@cyens.org.cy

Marios S. Pattichis
Department of Electrical and Computer Engineering
University of New Mexico
 Albuquerque, NM 87131-0001, USA
 pattichi@unm.edu

Efthymou Kyriacou
Department of Electrical Eng., Computer Engineering and Informatics,
Cyprus University of Technology
 Limassol, Cyprus
 efthymou.kyriacou@cut.ac.cy

Andreas S. Panayides
VIDEOMICS Group
CYENS Centre of Excellence
 Nicosia, Cyprus
 a.panayides@cyens.org.cy

Abstract—360° video streaming is one of the prevalent communication technologies for enhancing user experience and has thus seen widespread adoption in virtual and mixed reality applications. However, delivering content at scale while securing the quality of wirelessly communicated 360° videos in real-time poses significant challenges. 360° videos come in ultra-high definition, necessitate unprecedented bitrate demands and involve high encoding complexity. The time-varying nature of underlying wireless channels further introduces a destabilizing factor, calling for video systems to seamlessly adjust to varying bandwidth throughput to maintain adequate quality of service and experience. To address this issue, in this study, we have developed a multi-objective optimization framework for real-time video encoding adaptation. The objective is to optimize both video quality and encoding efficiency while minimizing the required bitrate, subject to real-time application constraints. To achieve this, we relied on generating (offline) precise forward prediction models of video quality, bitrate demands, and encoding time, that can be used to select the optimum encoding configuration in real-time. To validate our methods, we implemented an adaptive video encoding controller, and ran emulations employing actual network traces from 5G mobile video streaming scenarios, using the popular open-source x264 and x265 codecs for video encoding. A dataset of 4K omnidirectional videos at 30 frames per second was used.

Keywords—Adaptive Video Streaming, 360° videos, Video compression, HEVC, H264, 360° Video Streaming

I. INTRODUCTION

The ongoing hype in virtual reality (VR) and 360° video applications has triggered significant research interest, fueling the development of novel video streaming methods. Panoramic videos, offering a vast 360°×180° field of view (FoV) are pivotal in VR advancements [1]. Users, equipped with head-mounted displays (HMDs), can immerse and interact in a visual experience in a spherical panorama [2], seamlessly navigating through the omnidirectional video by simply moving their heads toward the desired directions [3]. At the same time, the prevalence of 360° video content has triggered new challenges in both the processing and streaming domains. To achieve high fidelity and prevent motion sickness in 360° video rendering, ultra-high definition (UHD) resolutions (e.g., 4K, 8K) and frame rates, are essential [1][4]. As a result, in addition to the

already heavy demands on storage, streaming services need to address the associated high bitrate demands and tight latency requirements [5]. In that sequence, efficient video compression and transmission methods are invaluable [6] [7].

5G (and beyond) wireless networks promise improved connectivity with significantly higher data rate and lower latency compared to 4G systems, driven by applications like social media, gaming, and augmented/virtual reality, while specifically supporting the breadth of IoT applications [8][9]. Predictions suggest a significant increase in connected IoT devices by 2025 [10]. In fact, the surge of connected devices associated with multimedia traffic, and particularly video, is a key factor pushing towards 5G networks' adoption. However, securing the quality of real-time, wireless 360° video streaming applications is challenging. The complexity arises from the dynamic nature of the underlying communication channels, which tend to fluctuate over time. Consequently, video systems must seamlessly adjust to the varying, instantaneous bandwidth throughput of the wireless medium at a given time, while maintaining the perceptual quality of the communicated videos.

In this study, we propose a multi-objective optimization approach that jointly maximizes a communicated video's quality and encoding rate while minimizing the associated bitrate demands, subject to time-varying, real-time application constraints. For that purpose, an adaptive video encoding controller is implemented that uses offline-generated forward prediction models for each of the optimization (constraint) objectives, namely video quality, bitrate demands, and encoding rate, resulting in the best possible encoding setup subject to the real-time constraints (objectives).

In Section II, we provide a brief overview of the state-of-the-art approaches in 360° video streaming. In Section III, we introduce the techniques used to develop forward prediction models, followed by the implementation of the multi-objective optimization framework aimed at achieving real-time video encoding adaptation to varying application and infrastructure specific constraints. Then, in Section IV we provide the experimental evaluation results along with the discussion of the key findings. Finally, in Section V we summarize the concluding remarks.



Fig. 1. 360° video image examples, Video Resolution: 4069 x 2048, Frame Rate: 40 fps, Projection Type: Equirectangular projection

II. RELATED WORK

The unprecedented interest and potential of 360° video applications is evident via an increasingly dense body of literature, especially over the past 5 years. Different methods have been proposed, each addressing unique aspects and characteristics of 360° videos, to optimize available resources and trigger wider adoption. Tile-based 360° video streaming linked with viewport prediction is a key concept for equirectangular projection (ERP)-based video format. The goal is to prioritize the user's FoV over the background, less important regions [3][11]. Assigning higher bitrate budget to the user's FoV has been found to increase the overall quality of experience (QoE), for both unicast and multicast scenarios. In the latter case, network-assisted techniques investigate the sharing of high-quality tile-based FoV regions and the caching of these video regions closer to the user [12]. At the same time, new or customized objective video quality assessment (VQA) metrics have been proposed that consider the FoV in the final (weighted) score, to achieve higher correlation to perceptual ratings [8]. In that context, a significant number of studies propose various viewport (i.e., FoV) prediction methods, based on historical viewport values captured via HMDs, using models like long short-term memory (LSTM) and deep reinforcement learning [11][13][14][15]. Deep learning models for automated region(s)-of-interest (ROIs) identification and translation to viewport are also gaining attention [16].

For real-time streaming services, an adaptation decision is considered at pre-defined intervals to maximize the user's QoE and a session's Quality of Service (QoS). In both 2D and 360° video communication systems, adaptive video streaming standards such as MPEG-DASH and HLS are used, primarily via the use of pre-constructed bitrate ladders. The latter is true for the proposed 360° video streaming approaches described earlier, that encompass customized bitrate ladders that consider and encode the user's viewport in higher quality [17]. Moreover, the use of base encoding layers and enhancement viewport layers, in a concept like scalable video coding is a popular approach. As a result, switching between pre-encoded states essentially limits the optimization process to a simple selection of the available encoding(s) that match the underlying bandwidth. To address this issue, more sophisticated methods have been proposed that utilize multi-objective optimization methods to examine and maximize additional objectives that go beyond the traditional rate-distortion optimization [18][19]. Such examples, include consumed encoding energy, dynamic viewport tiles selection and decisions based on multi-user video quality optimization as described above, but also self-organized network infrastructures [20] [21].

This study extends the multi-objective optimization approach developed by our group for conventional and medical 2D video transmission [18][19]. It jointly considers the

optimization of video quality, bitrate demands, and encoding rate. The approach is codec-agnostic, in the sense that it is applicable using any video codec and is underlying wireless network independent. Importantly, the proposed framework is complementary to other approaches in literature. In fact, ongoing work is tasked to investigate its effectiveness for tile-based video streaming using viewport prediction methods.

III. METHODOLOGY

In this section, we provide a detailed description of our proposed approach for adaptive video delivery of 360° video. Firstly, the methodology relies on an offline training phase, where 360° video instances involving different video encoding configurations are processed to derive objective forward prediction models for video quality, bitrate demands, and encoding time. Both the H.264/AVC [22] and H.265/HEVC [23] video compression standards are used for this purpose. Figure 1 captures a subset of the dataset's video content diversity. Then, the approach focuses on implementing an adaptive video encoding controller (see Fig. 3) that uses the generated forward prediction models and multi-objective optimization to materialize adaptive 360° video streaming, by triggering the encoding setup that maximizes the communicated video's quality (video quality objective), while matching the available bandwidth of the underlying wireless channel (bitrate demands constraint), in real-time (encoding time constraint).

A. Dataset

For validating the proposed approach, a dataset comprised of 5 omnidirectional videos of diverse content (see Fig. 1) was used, with 4K resolution (4096x2048 pixels), at 30 fps and a ten-second duration [24]. The raw videos were organized in 25 distinct video segments of 2 seconds each, using the ERP format.

B. Adaptive Video Delivery using Multi-objective Optimization

1) Video Compression Experimental Setup

In this study, we use open-source x264 and x265 [25][26] implementations that facilitate encoding optimizations enabling real-time encoding performance, being orders of magnitude faster than the JM and HM reference implementations of H.264/AVC [22] and HEVC/H.265 [23] video compression standards, respectively [18]. We consider different compression levels by adjusting the quantization parameter (QP), ranging from 16 to 37 with a step size of 1. The medium preset is used, which invokes the main profile for both codecs. As a result, for each codec used, we generated a total of 550 video segment instances (5 videos x 5 segments each x 22 different QPs).

2) Forward Prediction Models Generation

Forward prediction models are critical for implementing real-time adaptive video delivery using multi-objective optimization. The offline process for generating these models is depicted in the upper part of Figure 2. For each optimization objective, namely video quality, bitrate demands, and encoding frame rate (i.e. encoding time), a corresponding model is generated. For this purpose, a dense encoding space is needed, as tabulated in Table I. This space provides the critical mass of compressed video's outcome characteristics, that allows to investigate linear model estimation, which encompasses logistic regression, to derive accurate forward prediction models per optimization objective. This process underpins lightweight models, whose inverse solving is not computationally intensive, and hence qualifies for real-time streaming, provided that adequate accuracy is achieved.

TABLE I. VIDEO ENCODING CONFIGURATIONS

	Forward Prediction Models
Codecs	x264, x265
Preset	Medium
QP Range	16-37
Total configurations per codec	22
Total configurations	44

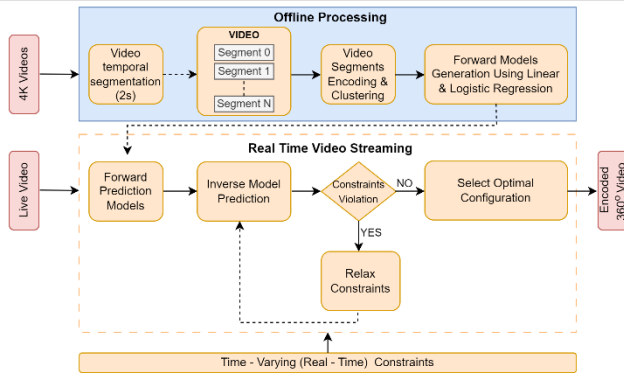


Fig. 2. Adaptive Video Encoding And Simulator Diagram

During the offline process, a dataset comprising five 360° omnidirectional videos with 4K resolution (4096x2048 pixels) at 30 frames per second was utilized. These videos were organized into 25 distinct video segments, each one with a 2 seconds duration. Each segment was then encoded using the libx264 and libx265 ffmpeg libraries, and the resulting encodings were employed to develop forward prediction models using linear and logistic regression, following video segments' clustering. During real-time simulation, inverse model prediction is employed to select the optimal encoding configuration that meets the application's constraints at each encoding adaptation interval.

a) Content-based Adaptation

Different video sequences have different, unique content characteristics. Some could be rich in spatial content while others could be rich in the temporal domain, involving high motion scenes. This variability can also be observed at an intra-video level. Thus, each video involves different spatiotemporal complexity, which is in turn translated into encoding complexity, resulting in diverse bitrate demands and encoding time. To address this issue, we propose a clustering approach to enhance the produced models' accuracy. Specifically, we apply

the k-means clustering method to organize the 25 segments of the videos into two distinct clusters. As shown in Figure 4, all segments of three of the videos belong in a single cluster, while video chunks of two of the videos are assigned to both clusters. A video's 5 constituent video segments are outlined with a red line while the two distinct clusters appear in yellow and purple, respectively. Based on this observation, we generate forward prediction models for each of the two clusters, aiming to minimize the overall prediction error during real-time video streaming sessions.

Algorithm 1 Pseudocode for Buffer Initialization

```

BufferTarget ← BufferSize × BufferPercentage
encodingConfiguration ← SelectedEncodingConfiguration()
while bufferFillingLevel ≤ BufferTarget do
    bandwidth ← ReadBandwidthFromNetworkTraces()
    if bandwidth ≠ NULL then
        compute BufferFillingLevel, segmentSize, availableBytes
    end if
    simulationTime ← simulationTime + 1
end while
return BufferFillingLevel, SimulationTime

```

Algorithm 2 Pseudocode for Selected Encoding Configuration

```

Bandwidth ← ReadBandwidthFromNetworkTraces()
FairnessSignal ← Bandwidth × BwAdjWeight
if BufferFillingLevel ≤ BufferMin then
    compute inputBandwidth
    return encodingConfiguration = maxSSIM(inputbandwidth, FPS)
else
    while True do
        compute inputBandwidth, encodingConfiguration,
        compute EstimatedDownloadTime, BwAdjWeight
        if encodingConfiguration.Bitrte ≤ BufferMin OR
        EstBufferFillevel > BufferMin then
            break
        end if
    end while
end if
return encodingConfiguration

```

Algorithm 3 Pseudocode for Adaptive Streaming

```

while (segments_to_encode > 0) do
    if (simulationTime == adaptation_interval) then
        inTransitSeg ← originalValue
        encodingConfiguration ← SelectedEncodingConfiguration()
    end if
    bandwidth ← readBandwidthFromNetworkTraces()
    if (bandwidth ≠ 0 and inTransitSeg ≠ 0) then
        compute downloadBytes, remainingSegBytes,
        compute BufferFillingLevel, InTransitSeg
        update simulationTime
    else
        if (BufferFillingLevel - 1 > 0) then
            BufferFillingLevel ← BufferFillingLevel - 1
            simulationTime ← simulationTime + 1
        else
            initializeBuffer()
            simulationTime ← simulationTime + bufferingTime
        end if
    end if
end while

```

Fig. 3. Overview of the Adaptive Video Streaming Simulator

Algorithm 1: *Buffer Initialization* occurs at the onset of every video streaming session or following a buffer drainage event.

Algorithm 2: *Encoding Configuration* is selected dynamically based on the real-time bandwidth readings extracted from the 5G network traces.

Algorithm 3: *Adaptive Video Streaming* simulator implementation.

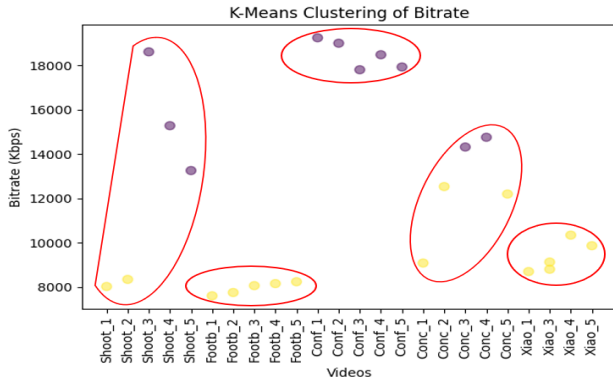


Fig. 4. K-means clustering of video segments, illustrating the division into two clusters (yellow and purple) based on the bitrate demands for QP of 22. The red outline depicts the segments belonging to the same video.

3) Multi-objective Optimization Framework

The approach here is to maximize video quality (denoted as VQ) measured in terms of Structure Similarity Index (SSIM) [27], minimize bitrate demands (denoted as B), and maximize encoding efficiency (or minimize encoding time), captured in encoding frames per second (denoted as FPS). Overall, the approach is depicted in the following equation:

$$MAX(VQ(SSIM), -Bitrate(B), FPS) \quad (1)$$

Based on (1), several multi-objective optimization modes can be defined [18][19]. In this study, the *Maximum Video Quality Mode* is used. The goal is to maximize the communicated video's quality, while adhering to onstraints on bandwidth availability imposed by the wireless infrastructure and encoding frames per second, imposed by the video's framerate. Hence, maximum video quality mode (*maxSSIM*) is characterized by the following constraints:

$$Bandwidth\ constraint: B \leq B_{max} \quad (2)$$

$$Frame\ rate\ constraint: FPS \geq FPS_{min} \quad (3)$$

C. Adaptive Video Encoding and Streaming Simulator

An adaptive video streaming simulator has been developed in Python to evaluate the performance of the proposed adaptive video encoding and control mechanisms under realistic wireless communication scenarios. The simulator parses real-life wireless network video traces to extract the instantaneous bitrate at any given point in time. The wireless network video traces comprise of 5G traces collected from two distinct mobility patterns, namely Static (indoors) and Driving (car). The car mobility scenario involved driving through city and suburban areas during the morning and evening hours. Experiments were conducted between 2019-11-20 and 2020-01-06, while the streaming video content was initiated from Amazon Prime services [9].

The adaptive streaming simulator uses a video buffer and dynamically adjusts video quality based on the available bandwidth, adhering to the logic of adaptive HTTP streaming standards [28]. Figure 3 summarizes the adaptive video simulator's algorithmic approach in pseudocode format. Algorithm 3 delineates the implementation philosophy of the video streaming session. In the setup phase of the streaming process, the video buffer is populated up to a predefined target

level to prevent video stalling and premature depletion of the buffer, as detailed in Algorithm 1 of Figure 3. During buffer initialization, the buffer is filled using a single encoding configuration to minimize the video streaming session's launch delay. Once the video starts at the receiver's end, and thereafter, the adaptive video simulator encodes successive video segments considering the available bandwidth and encoding rate constraints, aiming to select the best encoding configuration based on forward prediction models that maximizes video quality. The process involves making adaptation decisions every 2 seconds, known as the *adaptation interval* and is depicted in Algorithm 2 of Fig. 3.

TABLE II. FORWARD MODELS ADJUSTED R^2 MEDIAN VALUES

		Log (Bitrate)	Log (SSIM)	Log (FPS)
x264	Cluster 1	0.963	0.811	0.806
	Cluster 2	0.933	0.878	0.719
x265	Cluster 1	0.963	0.81	0.778
	Cluster 2	0.95	0.868	0.724

1) Adaptive Video Streaming Simulator Properties

Additional functionalities have been integrated into the adaptive video streaming procedure to enhance the effective utilization of the buffer, thereby improving the simulator's generalizability and scalability. Firstly, defining the *in-transit* value allows the simulator to download multiple segments simultaneously within a single simulation second. This functionality resembles a realistic approach concerning video segments' download subject to bandwidth availability that can extend over multiple simulation seconds. Secondly, implementing rules that aim to minimize buffer drainage incidents. Specifically, when the buffer level falls below a certain critical threshold during low-bandwidth conditions (i.e., less than 40%), the adaptation control algorithm automatically switches to a low-bitrate video encoding configuration, to facilitate the timely buffer fill-up.

IV. RESULTS

In this section, we provide adaptive video encoding and delivery, experimental evaluation results. The experiments were performed on a Windows 10 64-bit HP OMEN Tower 10 (v.22 H2), with 12th Gen Intel(R) Core (TM) i9-12900K (16 cores, 3.20 GHz).

1) Forward Prediction Models Generation

The logistic linear models displayed below as a function of the quantization parameter were determined to be the best and most resilient, in terms of the achieved adjusted R squared values and overall prediction error minimization:

$$\log(Bitrate) = a1 * QP + b1 \quad (4)$$

$$\log(SSIM) = a2 * QP + b2 \quad (5)$$

$$\log(FPS) = a3 * QP + b3 \quad (6)$$

Here, $a1$, $a2$, $a3$ represent the QP coefficients, while $b1$, $b2$, $b3$ denote constants. The equations (4), (5), and (6) are utilized to differentiate between the Bitrate, SSIM and FPS objectives, respectively.

Table II demonstrates the results for the median values of the adjusted R^2 per optimization objective, for both the x264 and x265 codecs, and the two clusters. The adjusted R square was calculated using the leave-one-out 5-fold cross validation method. In all cases, adjusted R^2 values are higher than 0.72. These high values demonstrate the robustness of the derived forward prediction models.

B. Average Prediction Error under Realistic Conditions

Figure 5 demonstrates the percentage (%) prediction error for (a) bitrate demands and (b) video quality, after averaging the difference between the predicted values using the forward prediction models and the actual video encoding results, over all simulation runs, encompassing the total number of wireless network video traces, for the driving mobility scenario. The static mobility scenario yields similar results but is not depicted here due to space constraints. For each graph, the average prediction error per investigated video codec is given, namely x264 and x265, for in-transit values of 2 and 4 (in-transit value of 3 achieves similar results to in-transit value of 4, as shown later in Figure 6), respectively. Moreover, results are given for the two distinct scenarios involving content-adaptation (i.e., different forward prediction models used per the two clusters highlighted in Figure 4) and no content-adaptation (i.e., one set of forward prediction models). Lastly, the average percentage

prediction error is computed based only on the QP range values {12-37} that the models were trained upon that further portray the typical video streaming content compression levels [29][30].

Three key observations can be deduced by the bitrate demands prediction error results appearing in Figure 5 (a). First, the x265 codec is associated with a lower, approximately half average prediction error compared to its predecessor, the x264 codec. Second, the content adaptation approach reduced the prediction error for both codecs. Here, it is important to highlight that switching between the generated forward prediction models per identified cluster(s) is considered every 10 seconds, with a switch taking place to the forward prediction model set that minimizes the prediction error. In that context, the depicted results agree with the afore-described methodology. Third, there is no significant difference between the in-transit value, as expected.

In terms of video quality, the average error measured using the SSIM index is less than 1% for all investigated models. The latter emphasizes the accuracy and robustness of the video quality prediction models. Like bitrate demands, the average SSIM prediction error decreases when using the content adaptation approach. However, this does not constitute a statistically significant reduction in the present experiments.

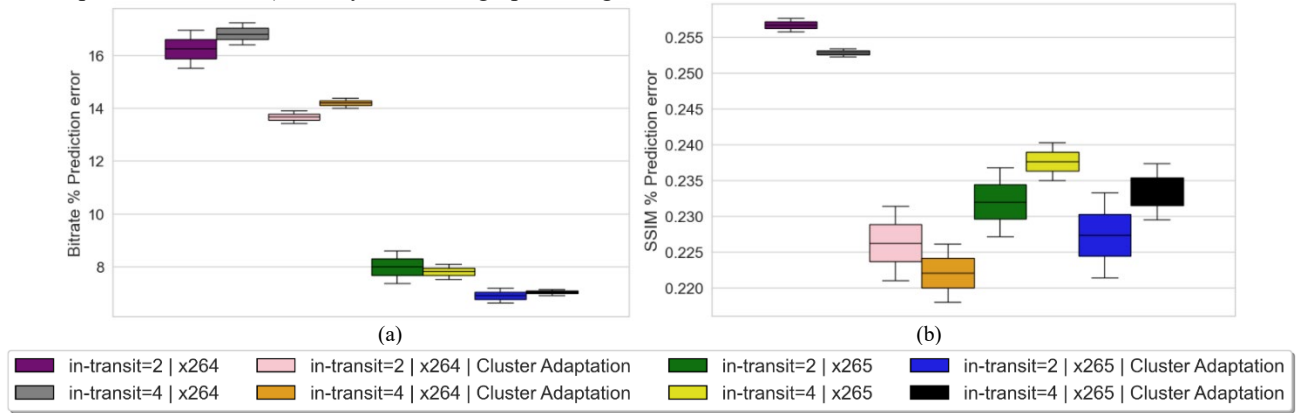


Fig. 5. Boxplots (a), (b) depict the percentage (%) prediction error for (a) bitrate demands and (b) video quality, based on the QP range values {12-37}, for Amazon Prime network video traces during the driving mobility. Figures illustrate the prediction errors for in-transit values of {2 and 4} while utilizing the maximum video quality mode with and without cluster adaptation.

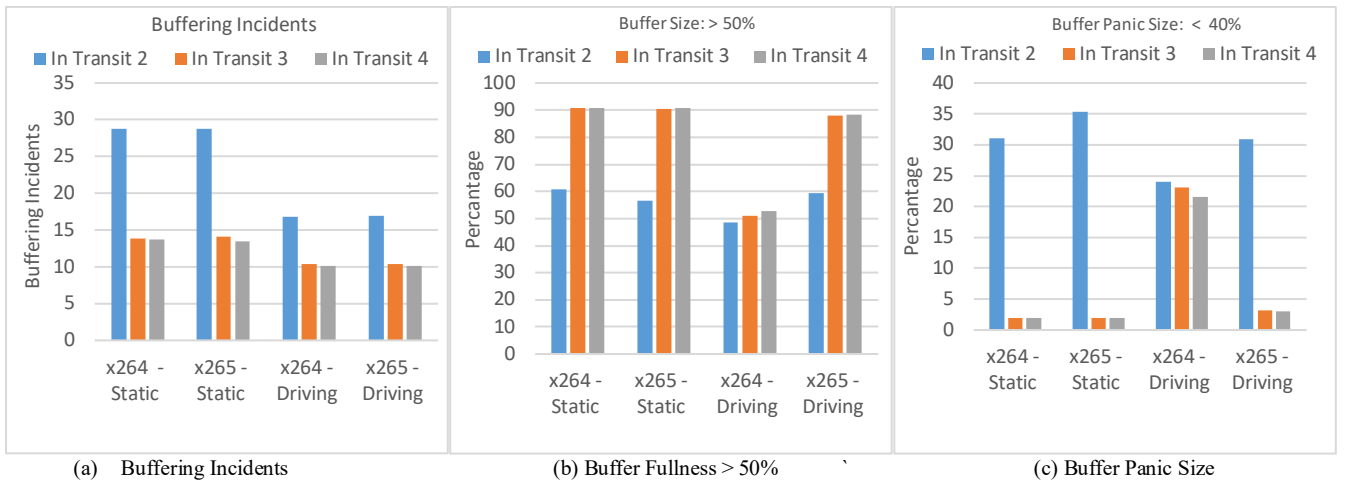


Fig. 6. 360° Video Streaming QoS Buffer Statistics for (a) Buffering Incidents, (b) Buffer Fullness >50%, and (b) Buffer Panic Size (buffer fullness <40%).

C. 360° Video Streaming Quality of Service Metrics

The results presented in Figure 6 document a notable enhancement in all Quality of Service (QoS) metrics when the number of in-transit segments is increased from 2 to {3 and 4}. The latter option defines the maximum number of video segments (in seconds) that can be downloaded and added to the buffer within a single adaptation interval. Specifically, when the in-transit value was set to 2, the buffer fullness remained below the buffer panic threshold of 4 seconds for a period between 24%-35%, depending on the investigated scenarios. On the contrary, with in-transit set to 3 and 4, the maximum time percentage near the rebuffering limit was significantly lower, in the range of 1%-3%. Similarly, for an in-transit value of 2, the buffer fullness exceeded the 50% threshold for only 47%-49% of the total simulation time for all scenarios, whereas this percentage was notably raised when in-transit value was set to 3 and 4, reaching up to 91% and 88% for the static and driving mobility scenarios, respectively. With respect to the number of buffering incidents that cause a video to stall due to buffer drainage, continuing until an adequate buffer threshold is restored, again, there is a significant reduction when the in-transit values are increased, as we can see in Figure 6. Based on the above observations, we can conclude that the duration of downloaded segments may exceed the adaptation interval. This approach can enhance the user's Quality of Experience by allowing the simulator to achieve high buffer fullness, thereby helping to mitigate the number of video stalling incidents.

V. CONCLUSIONS

This study proposed an adaptive 360° video encoding and streaming approach that utilizes multi-objective optimization to address real time varying constraints. Using lightweight forward prediction models per optimization objective, the proposed approach can be used to achieve effective and efficient adaptation, with low prediction error. Moreover, for unicast applications, the depicted methods are far more elastic and responsive compared to adaptive HTTP streaming methods that are limited to the pre-encoded bitrate ladder. Currently, ongoing work investigates the applicability of the proposed approach for multicast applications. Using the maximum video quality mode with content adaptation support resulted in better prediction results, reducing the average prediction errors for all the investigated parameters. Moreover, the suggested techniques enhance the user's QoE and session's QoS, associated with reduced video stalling and buffering incidents.

REFERENCES

- [1] Z. Chen, Y. Li, and Y. Zhang, "Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation," *Signal Processing*, vol. 146, pp. 66–78, 2018.
- [2] M. Wien, J. M. Boyce, T. Stockhammer, and W.-H. Peng, "Standardization status of immersive video coding," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2019.
- [3] M. Mahmoud, S. Rizou, A. S. Panayides, "A Review of Deep Learning Solutions in 360° Video Streaming", 2023 12th International Conference on Modern Circuits and Systems Technologies (MOCASST), Athens, Greece, 2023, pp. 1-4, Doi: <https://doi.org/10.1109/MOCASST57943.2023.10176729>
- [4] J. Nightingale et al. 2018. 5G-QoE: QoE Modelling for Ultra-HD Video Streaming in 5G Networks. *IEEE Transactions on Broadcasting* 64, 2 (June 2018), 621–634. Doi: <https://doi.org/10.1109/TBC.2018.2816786>
- [5] Bastug, E., Bennis, M., Médard, M., & Debbah, M. (2017). "Toward interconnected virtual reality: Opportunities, challenges, and enablers", *IEEE Communications Magazine*, 55(6), 110-117.
- [6] A. S. Panayides, et al., "The Battle of the Video Codecs in the Healthcare Domain - A Comparative Performance Evaluation Study Leveraging VVC and AV1," in *IEEE Access*, vol. 8, pp. 11469-11481, 2020, doi: 10.1109/ACCESS.2020.2965325.
- [7] I. Valiandi, et al. "A Comparative Performance Assessment of Different Video Codecs." In *Lecture Notes in Computer Science*, In: Tsapatsoulis, N., et al. *Computer Analysis of Images and Patterns. CAIP 2023. Lecture Notes in Computer Science*, Springer, pp. 265–275. Doi: https://doi.org/10.1007/978-3-031-44240-7_26
- [8] Xu, M.; Li, C.; Zhang, S.; Callet, P.L. "State-of-the-Art in 360° Video/Image Processing: Perception, Assessment and Compression", *IEEE J. Sel. Top. Signal Process.* 2020, 14, 5–26.
- [9] D. Raca, D. Leahy, et al. "Beyond throughput, the next generation: a 5G dataset with channel and context metrics", In *Proceedings of the 11th ACM Multimedia Systems Conference (MMSys '20)*. Association for Computing Machinery, New York, NY, USA, 303–308, 2020. Doi: <https://doi.org/10.1145/3339825.3394938>
- [10] Alam, Tanweer "A Reliable Communication Framework and Its Use in Internet of Things (IoT)", 2018, <https://ssrn.com/abstract=3619450>
- [11] Yaqoob, A. et al., "Dynamic viewport selection-based prioritized bitrate adaptation for tile-based 360° video streaming.", *IEEE Access*, 10, 29377–29392. Doi: <https://doi.org/10.1109/access.2022.3157339>, 2022.
- [12] M. Mahmoud et al., "A Survey on Optimizing Mobile Delivery of 360° Videos: Edge Caching and Multicasting," in *IEEE Access*, vol. 11, pp. 68925–68942, 2023, doi: 10.1109/ACCESS.2023.3292335.
- [13] S. Park, et al. "Adaptive Streaming of 360-Degree Videos with Reinforcement Learning." Doi: 1838-1847. 10.1109/WACV48630.2021.00188, 2021
- [14] H. Viet, et al., "Scalable and Resilient 360-degree-Video Adaptive Streaming over HTTP/2 against Sudden Network Drops", 10.36227/techrxiv.21780704.v3., 2023
- [15] J. Zhang, Y. Zhong, et al., "A 360° Video Adaptive Streaming Scheme Based on Multiple Video Qualities," *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, Leicester, UK, 2020, pp. 402–407, doi: 10.1109/UCC48980.2020.00063
- [16] Yaqoob, A., et al. "A Edge Caching on adaptive 360 video streaming: Solutions, challenges and opportunities". *IEEE Communications Surveys & Tutorials*, 22(4), 2801–2838.
- [17] Fan, C. L., et al. "A survey on 360 video streaming: Acquisition, transmission, and display." *Acm Computing Surveys* 52(4), 1–36, 019
- [18] G. Esakki, et al., "Adaptive Video Encoding for Different Video Codecs," in *IEEE Access*, vol. 9, pp.68720–68736, 2021, Doi: <https://doi.org/10.1109/ACCESS.2021.3077313>
- [19] Z. C. Antoniou, et al., "Real-Time Adaptation to Time-Varying Constraints for Medical Video Communications," in *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, pp. 1177–1188, July 2018. doi: 10.1109/JBHI.2017.2726180
- [20] L. Zhao, et al., "Adaptive Streaming of 360 Videos With Perfect, Imperfect, and Unknown FoV Viewing Probabilities in Wireless Networks," in *IEEE Transactions on Image Processing*, vol. 30, pp. 7744–7759, 2021, doi: 10.1109/TIP.2021.3099741.
- [21] Duanmu, F., et al. "View Direction and Bandwidth Adaptive 360 Degree Video Streaming using a Two-Tier System.", New York University, Tandon School of Engineering, Brooklyn, NY 11201, USA, 2024
- [22] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [23] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [24] Chen Li, Mai Xu, et al. "Bridge the Gap Between VQA and Human Behavior on Omnidirectional Video", In *Proceedings of the 26th ACM international conference on Multimedia (MM '18)*. Association for Computing Machinery, New York, NY, USA, 932–940, 2018, Doi: <https://doi.org/10.1145/3240508.3240581>

- [25] FFmpeg's x264 build, libx264, <https://trac.ffmpeg.org/wiki/Encode/H.264>
- [26] FFmpeg's x265 build, libx265, <https://trac.ffmpeg.org/wiki/Encode/H.265>
- [27] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error measurement to structural similarity, *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004
- [28] ISO/IEC 23009-1, Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part1: Media presentation description and segment formats, 5th edition, 08/2022.
- [29] Wowza Video Platform, "Video Bitrate Recommendations by Platform (YouTube)", Available online: <https://www.wowza.com/blog/what-is-video-bitrate-and-what-bitrate-should-you-use>, Last Accessed: March 2024
- [30] IBM Video Streaming Support Center, "Recommended Encoding Settings", Available Online: <https://support.video.ibm.com/hc/en-us/articles/207852117-Internet-connection-and-recommended-encoding-settings>, Last Accessed: March 2024