Bidirectional Buffer-Constrained Bitrate Adaptation for layered 360-degree Video Streaming

***Abstract*—Viewport adaptive streaming can efficiently reduce data transmission while maintaining the expected user's Quality of Experience(QoE), which is a enabling technology for online 360-degree video applications under constrained network bandwidth. A critical challenge is how to to balance the video rebuffering and quality under different network conditions, considering the imperfect viewport prediction and constrained buffer size at the user side. However, existing works tends to optimize QoE represented by the weighted sum of rebuffering and quality. In this work, we aim to maximize the expected video quality while guarantee the long-term average buffer remains within a specified range. We propose the Lyapunov Stable Point Optimization(LSPO), a Lyapunov optimization variant. LSPO achieves a near-optimal time-averaged QoE within an $O(1/V)$ margin, while also upholding bidirectional buffer constraints. A buffer-based algorithm is then proposed and evaluated based on real-trace data. The results demonstrate superior performance in video quality and reduced rebuffering probability compared to existing algorithms.**

***Index Terms*—layered 360° video streaming, Lyapunov Stable Point Optimization, bidirectional buffer-constrained**

I. Introduction

The global virtual reality (VR) market, valued at $28 billion in 2022, is expected to grow at a compound annual growth rate of 13.8% from 2023 to 2030 [1]. As a pivotal VR application, 360-degree video offers an immersive viewing experience, significantly transforming our interaction with virtual environments. Nonetheless, the file sizes of 360-degree videos are about 4 to 6 times larger than those of standard videos, presenting substantial challenges for network transmission during online streaming. For instance, 4K 360-degree video streams on YouTube often experience rebuffering, which can constitute up to 33% of total playback time [2]. Thus, improving 360-degree video streaming performance under bandwidth constraints has become a critical area of research.

Considering users only view a portion of the 360-degree video, known as the viewport, an effective method to enhance user experience is to transmit only the viewport rather than the full video frame [3]. However, the accuracy of viewport prediction tends to decline as the forecast interval extends, leading to a standard buffer duration of merely 1-3 seconds [4]. This short buffer renders the video bitrate highly vulnerable to fluctuations in network bandwidth, and increases the risk of video rebuffering. Layered 360-degree video streaming [5] is a promising approach which prefetches the entire video frame at the lowest bitrate (i.e., the base layer) and downloads the viewport at a higher bitrate (i.e., the enhancement layer) shortly before playback. This method utilizes Scalable Video Coding(SVC) [6] to allow the enhancement layer to decode using the base layer. To avoid interruptions in streaming, it's crucial to prevent the base layer buffer from under-running Moreover, the buffer capacity is usually limited at the user side. For example, conventional video streaming practices typically involve a 20-second buffer [7], hence introducing bidirectional buffer constraints.

Existing methods typically prioritize buffer constraints, allowing the bitrate to increases linearly with buffer length, but they often neglect the relationship between video quality and bitrate. Some approaches opt for a weighted sum function that merges video rebuffering and quality metrics. However, equating video rebuffering with quality presents a significant challenge due to their inherently incomparable nature, complicating the task of setting appropriate weights. Consequently, a crucial question arises: *How can we dynamically adjust the bitrate within bidirectional buffer constrained to optimize video quality?* Network bandwidth is dynamic in practice and cannot be accurately predicted in the long term, posing great challenges to the design of online bitrate adaptation schemes. Lyapunov optimzation offers a pathway to near-optimal solutions without full state knowledge [8]; yet it traditionally caters only to unidirectional buffer constraints. This observation prompts another critical inquiry: *How can we achieve long-term optimal decisions amidst uncertain bandwidth conditions?*

To tackle these challenges, we formulate bitrate adaptation as a problem of maximizing video quality, subject to bidirectional buffer constraints. We propose Lyapunov Stable Point Optimization(LSPO), a variant of Lyapunov optimization to respect bidirectional buffer constraints and secure near-optimal outcomes despite the lack of complete global information. Specifically, the buffer is represented as two virtual queues and a novel queue dynamic equation is introduced to separate virtual queues. We prove that our approach aligns with the principles of Lyapunov optimization. We developed a bitrate adaptation algorithm based on LSPO, which first recasts the stochastic control problem into a single-time-slot optimization issue, and then address it through a heuristic method. Empirically validated by real-world bandwidth dataset and user trajectories, our methodology surpasses existing algorithms in both video quality and rebuffering probability.

To summarize, this paper makes the following key contributions:

- We introduce a novel framework for 360-degree video streaing that leverages SVC and design a preemptive

layered transmission scheduling process that transitions from continuous-time to discrete-time scheduling.
- We propose LSPO, a variant of Lyapunov optimization [9]. LSPO ensures bidirectional long-term average buffer constraints, and achieves an $O(1/V)$ approximate solution to the optimal problem. Subsequently, we employ LSPO to design a bitrate adaptation algorithm.
- We conduct comprehensive simulations on 360° video to confirm the practicality of our algorithm. The experimental results, supported by public datasets, indicate that our method outperform existing algorithms in terms of both video quality and rebuffering probability.

II. Related Work

Viewport-based 360° video streaming: With bandwidth limitations posing a significant challenge, viewport-based 360-degree video streaming emerges as a strategic approach to minimize data transmission by focusing on the user's immediate field of view. This method consists of three core components: viewport prediction, bandwidth prediction, and adaptive bitrate algorithms. Viewport can be predicted based on the user's viewing trajectory and video content, including the trajectories of prime objects [10] and panoramic saliency [11]. Different viewport prediction methods are proposed by employing linear regression [12], LSTM [13], and ensemble learning [14]. Despite the method used, prediction accuracy decreases over time. Adaptive bitrate algorithms adjust the bitrate based on the viewport and buffer status to maximize QoE, which is usually modeled as an optimization problem and addressed by employing dynamic programming [15], reinforcement learning [16], beam search [14], and gradient descent [17]. However, due to issues with prediction accuracy, these methods rely on short buffers, thus making video bitrates susceptible to fluctuations in bandwidth. Almquit *et al.* [5] highlight a compromise between buffering for accuracy and avoiding rebuffering.

Layered 360° video streaming: Duanmu *et al.* propose a two-tier video streaming to address the limitations of viewport-based streaming [18]. Wherein the high-rate enhancement layer maintains a short buffer, while the low-rate base layer maintains a long buffer. If the base layer buffer length falls below the target buffer length, the client will download base layer chunks sequentially. A P-I (Proportional-Integral) controller manages the enhancement layer buffer when the buffer length exceeds the target. These methods prioritize buffer constraints while considering video quality enhancement as secondary. Sun *et al.* explore multipath transmission, utilizing WiFi for two-tier streaming and 5G for correction and retransmission [19]. Sun *et al.* considers the network bandwidth statistics and viewport prediction accuracy for optimal rate allocation between the enhance layer and base layer [20]. Zhang *et al.* propose a hierarchical buffer management approach [21], further refined by their subsequent work, which employed the *t* location-scale distribution for improved viewport prediction error [22]. However, these methods primarily focus on immediate optimizations, neglecting long-term playback quality.

Long-term optimal bitrate adaptation: Aiming at more stable under bandwidth flucations, Spiteri*et al.* [23] and Zhang *et al.* [24] propose buffer-based adaptive bitrate algorithms for traditional and 360-degree video streaming, respectively. These algorithms leverage Lyapunov optimization to attain near-optimal utility. The Lyapunov optimization [9] can provide an approximate optimal solution under the unidirectional buffer constraint, even without predicting the available network bandwidth. Therefore, to meet the bidirectional constraints of the buffer, these algorithms consider the playback time to total duration ration as a measure against buffer underflow. Subsequently, they optimize a linear weighted sum of this ratio and video quality. Linearly combining two quantities with different dimensions is challenging to adapt to all network conditions.

In response these challenges, we introduce an SVC-based 360-degree video layered transmission framework that incorporates a preemptive layered transmission scheduling process, transforming continuous-time scheduling into discrete-time scheduling. Contrasting with previous approaches that optimize linear weighted multi-objectives, We propose LSPO to ensure near-optimal QoE amidst buffer overflow and underflow constraints. By applying LSPO, we simplify the long-term optimization challenge into a single-slot optimization problem. We develop a heuristic online adaptive algorithm to solve the single-slot optimization problem.

III. System Model

This section delineates our SVC-based 360-degree video layered transmission framework, as depicted in Figure 1.

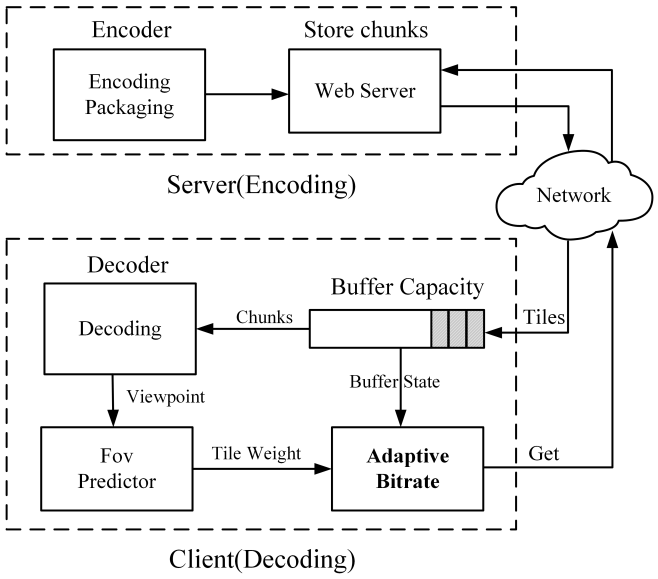

Fig. 1. The framework of the layered 360-degree video streaming.

Our framework integrates a server responsible for video encoding, storage, and handling HTTP requests from users. On the client side, a FoV predictor is employed to forecast the viewport. Leveraging the prediction results and current

buffer status, an adaptive bitrate algorithm is implemented to determine the download bitrate. The operation timeline is segmented into equal-length, non-overlapping time slots, each corresponding to the playback completion of a video chunk, indexed by t, where $t \in \{1, ..., T\}$.

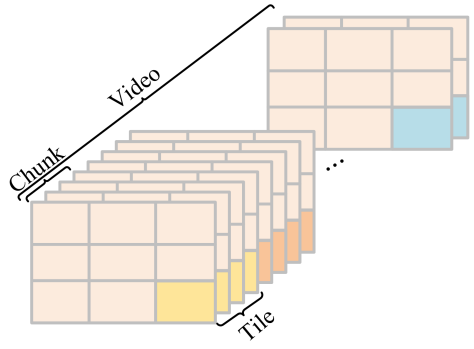

Fig. 2. Video is divided temporally into chunks, and each chunk is segmented spatially into tiles.

Server Model: The video content is divided into N chunks, indexed as $\{1, 2, ...n, ..., N\}$, with each chunk having a duration of τ seconds, as shown in Fig. 2. Each chunk is spatially divided into I tiles, indexed as $\{1, 2, ...i, ..., I\}$. By applying SVC, each tile is encoded into M different quality levels, with bitrate represented as $\{v_1, v_2, ..., v_m, ..., v_M\}$. The lowest quality level, v_1, constitutes the base layer, while higher quality levels form the enhancement layers.

Transmission Model: We adopt a generalized time-varying model for video transmission with the available bandwidth, $\omega(t)$, fluctuating over time. This model does not presuppose knowledge of bandwidth distribution.

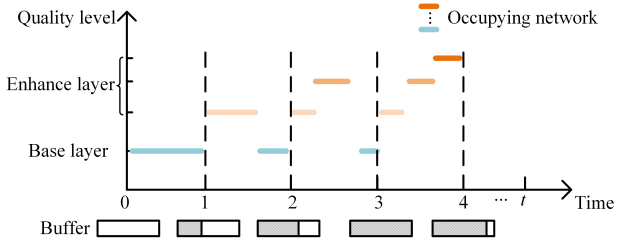

Fig. 3. Preemptive layered transmission time distribution.

Client Model: At the client end, a layered prefetching and downloading strategy is employed to mitigate video rebuffering and maintain quality amid varing bandwidth and viewport prediction accuracy. The base layer is fully prefetched irrespective of the viewport to ensure playback continuity. Conversely, the enhancement layer for the imminent chunk is downloaded based on current viewport prediction. The server persistently transmits the base layer. Upon receiving a request for an enhancement layer, the server gives it immediate priority. Each time slot triggers a client-side bitrate adaptation algorithm to request the enhancement layer from the server as shown in Fig. 3.

The buffer length in slot t is denoted by $B(t)$, measured in seconds. The viewport predictor predicts the probability, $w_{t,i}$, of viewing the tile i in time slot t, guiding the adaptive bitrate algorithm in selecting the download bitrate, $r_{t,i}$, for each tile.

IV. Problem Formulation

A. QoE Formulation

The QoE hinges on video quality and rebuffering. The video quality encompasses two components: expected viewport quality and expected viewport quality oscillation due to bitrate variability across adjacent tiles. Following the relationship outlined by Spiteri *et al.* [23], we assume a logarithmic correlation between Peak Signal-to-Noise Ratio (PSNR) and bitrate. The expected viewport quality is given by:

$$U^v(t) = \sum_{i=1}^{I} w_{t,i} log(\frac{r_{t,i}}{v_1}) \tag{1}$$

To mitigate visual discontinuities, we aim to minimize the bitrate difference between adjacent tiles, leading to the expression for expected viewport quality oscillation:

$$U^o(t) = \sum_{i=1}^{I} \sum_{j \in S(i)} \frac{w_{t,i} + w_{t,j}}{2} \frac{1}{16} log(|\frac{r_{t,i}}{r_{t,j}}|), \tag{2}$$

where the set $S(i)$ represents the eight tiles adjacent to one tile.

Therefore, the video quality can be formulated as follows:

$$U(t) = U^v(t) + \mu U^o(t), \tag{3}$$

where μ denotes the significance of quality oscillation in the overall video quality.

The base layer is prefetched and buffered. Reflecting the preemptive layered transmission aporoach illustrated in Fig. 3, the variation of buffer length can be modeled as an G/D/1 queue. The arrival and service rates, $a(t)$ and $b(t)$ for the queue are given by:

$$a(t) = \tau \frac{\omega_t - \sum_{i=1}^{I}(r_{t,i} - v_1)}{\sum_{i=1}^{I} v_1}, b(t) = \tau, \tag{4}$$

where $a(t)$ and $b(t)$ represent base layer downloading time and video consumption during time slot t, respectively.

The dynamic equation for the buffer can be represented as follows:

$$B(t+1) = max\{B(t) - \tau, 0\} + \tau \frac{\omega_t - \sum_{i=1}^{I}(r_{t,i} - v_1)}{\sum_{i=1}^{I} v_1} \tag{5}$$

The combined bitrate for all tiles within a given time slot is constrained not to exceed the available bandwidth:

$$\sum_{i=1}^{I} r_{t,i} \leq \omega_t \tag{6}$$

B. Problem

Given the inherent unpredictability of network bandwidth over the long term, it's impractical to assure that the buffer constraints will always be met. To navigate this challenge, we adopt a time-averaged buffer constraint approach, as suggested by Spiteri et al. [23], defined by:

$$B_{\min} \leq \frac{1}{T}\sum_{t=1}^{T} B(t) \leq B_{\max}, \tag{7}$$

where $B_{\max}$ and $B_{\min}$ respectively represent the upper and lower bounds for the long-term average buffer length.

We then formulate the optimization problem as Problem 1(P1).

$$\begin{aligned} (\text{P1}) \quad & \underset{\{r_{t,i}|t\in\{1,..,T\},i\in\{1,...,I\}\}}{\arg\max} \quad \frac{1}{T}\sum_{t=1}^{T} U(t) \\ & \text{subject to} \quad (6), (7) \end{aligned} \tag{8}$$

Lyapunov optimization [9] has been adopted to address similar problems of (P1) when only consider the $B_{\max}$ constraint, i.e., unidirectional queue constraint. To address this problem, we propose segmenting these constraints into two manageable unidirectional virtual queues by introducing a buffer's stable point, thus separating the buffer into regions where the constraints of $B_{\max}$ and $B_{\min}$ apply. This division, however, introduces interdependence between decisions affecting each side of the stable point. To counteract this issue, we refine our model with a new dynamic equation for the two virtual queues that maintains the i.i.d nature of arrival and service rates across time slots, effectively encapsulating the interdependence. This adaptation ensures the applicability of Lyapunov optimization to our revised model, culminating in the derivation of the LSPO to address Problem 1.

V. Online Adaptation Algorithm

First, we construct virtual queues to split the buffer and propose new queue dynamic equations to decouple the interdependence of these virtual queues. Secondly, we prove the existence of an optimal solution for Problem 1. We then propose LSPO and transform Problem 1 into a single time slot optimization problem. Finally, we present an online heuristic algorithm to solve the single time slot problem.

A. New queue dynamic equation

We set S as a stable point which satisfies $B_{\min} < S < B_{\max}$ and try to ensure buffer length does not deviate far from S, which is consistent to the bidirectional buffer constraint. We focus on the buffer length at the start of each time slot, which is used for bitrate decision making. As shown in Fig. 4, two virtual queues are constructed based on the stable point S. Virtual queue 1 extracts the buffer status when the buffer length exceeds S, with the timeline denoted by $1, 2, ..., T_1$. In addition, virtual queue 2 extracts the buffer status when the buffer length is below S, with the timeline denoted by $1, 2, ..., T_2$. Notice that $T_1 + T_2 = T$. The length of two virtual

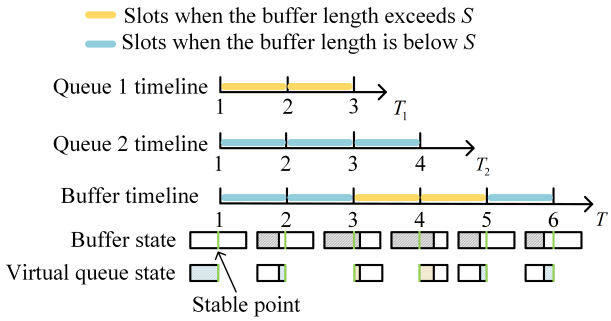

Fig. 4. Correlation between virtual queue and buffer $B(t)$

queues describe the distance between the buffer length and the stable point, given by

$$\begin{cases} Q_1(t) = B(t + n_1(t)) - S, & \text{if} B(t + n_1(t)) \geq S \\ Q_2(t) = S - B(t + n_2(t)), & \text{if} B(t + n_2(t)) < S \end{cases}, \tag{9}$$

where $Q_1(t)$ and $Q_2(t)$ represent the length of virtual queue 1 and virtual queue 2 at the beginning of time slot t, respectively, and $n_1(t)$ and $n_2(t)$ represent the duration extracted by virtual queue 1 and virtual queue 2, respectively, relative to the buffer.

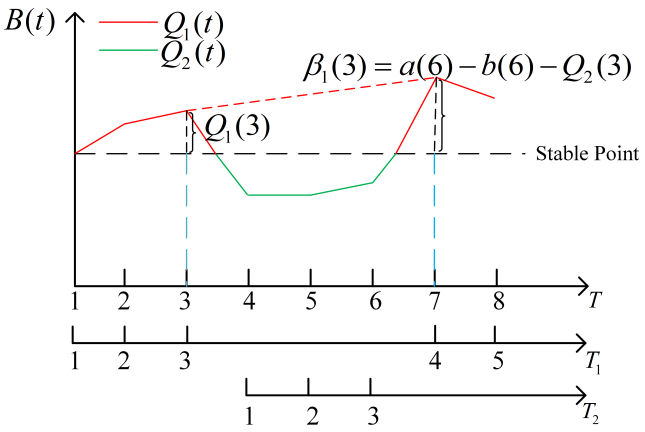

Fig. 5. Variation in buffer length over time.

As depicted in Fig.5, when the buffer fluctuates around the stable point(from time T = 3 to 7), the arrival and service rates of the virtual queue are not only related to the arrival and service rates of the buffer but are also influenced by the other virtual queue length. A necessary condition for the applicability of Lyapunov Optimization is that both the arrival and service rates must be independent withing each time slot. To achieve temporal independence, we divide the fluctuation around the stable point within a time slot into two steps. In the first step, the virtual queue is reduced to zero based on arrival and service rates that are independent of the time slot. In the second step, the virtual queue, having been reset to zero, is bounced back to a bounded value. The bounded variable is given by:

$$\beta_1(t) \le a(t+n_1(t)) - b(t+n_1(t)) \le C \tag{10}$$

$$\beta_2(t) \le b(t+n_2(t)) - a(t+n_2(t)) \le C, \tag{11}$$

where C represents the absolute value of the maximum difference between the arrival and service rates of the buffer. The arrival rate and service rate(which can be negative [9]) of the virtual queue are:

$$a_1(t) = 0, b_1(t) = b(t+n_1(t)) - a(t+n_1(t)) \tag{12}$$

$$a_2(t) = 0, b_2(t) = a(t+n_2(t)) - b(t+n_2(t)) \tag{13}$$

Based on the above description, the new queue dynamic equation[1] is:

$$\begin{aligned} Q_1(t+1) =& [Q_1(t) - b_1(t) < 0 ? \beta_1(t) \\ & : Q_1(t) - b_1(t)] + a_1(t) \end{aligned} \tag{14}$$

$$\begin{aligned} Q_2(t+1) =& [Q_2(t) - b_2(t) < 0 ? \beta_2(t) \\ & : Q_2(t) - b_2(t)] + a_2(t) \end{aligned} \tag{15}$$

Next, we will prove that Lyapunov optimization is compatible with the new dynamic equations. Specifically, the incorporation of queue rebound bounded variables $\beta_1(t)$ and $\beta_2(t)$ does not affect the bounded nature of the queue.

B. Existence of the Optimal Solution

In this subsection, we demonstrate the existence of an optimal solution that satisfies the rate stable condition for virtual queue 1 over the time slots from 1 to T_1.

Lemma 1: If the rebound variable is bounded, the sufficient and necessary condition for a queue following the new dynamic equation to be rate stable is that the average arrival rate is less than or equal to the average service rate:

$$\lim_{t->\infty} \frac{Q(t)}{t} = 0 \iff E[a(t)] \le E[b(t)] \tag{16}$$

Proof: see the appendix.

Assuming we have knowledge of all possible outcomes for network conditions and tile weight, represented by the set P, with p indicating a specific outcome withing the this set. $\Pr(p)$ represents the probability of each outcome occurring. Let $\alpha_1(t)$ represent the transmission decisions which refer to the bitrate of each download. Therefore, the arrival and service rates for each time slot are given by:

$$\begin{aligned} E[a_1(t)] &= \sum_{p\in P} \Pr(p) a_1(\alpha_1(t)) \\ E[b_1(t)] &= \sum_{p\in P} \Pr(p) b_1(\alpha_1(t)) \end{aligned} \tag{17}$$

We define $U_1^*(\epsilon_1)$ as the maximum average value that can be achieved by any P-only algorithm that makes virtual queue 1 rate stable. By applying **Lemma 1**, the value $U_1^*(\epsilon_1)$ can be computed by solving the following single-slot optimization problem:

$$\begin{aligned} \arg\max_{\alpha_1} \quad & U_1^* \triangleq \sum_{p\in P} \Pr(p) U(p, \alpha_1(t)) \\ \text{subject to} \quad & \sum_{p\in P} \Pr(p) a_1(\alpha_1(t)) \le \sum_{p\in P} \Pr(p) b_1(\alpha_1(t)) \\ & \sum_{p\in P} \Pr(p) = 1 \end{aligned} \tag{18}$$

Since the decision variables are discrete and finite, an optimal solution must exist. For each $0 \le \epsilon_1 \le \epsilon_{max}$, there is a P-only algorithm $\alpha_1^*(t)$ such that:

$$\begin{aligned} E[U(\alpha_1^*(t), P(t)] &= U_1^*(\epsilon_1) \\ E[a_1(\alpha_1^*(t), P(t)) - b_1(\alpha_1^*(t), P(t))] &= -\epsilon_1 \end{aligned} \tag{19}$$

ϵ_{max} is the maximum value of the mean difference between the average arrival and service rates.

C. Lyapunov Stable Point Optimization

In this subsection, we prove LSPO and, based on LSPO, transform (P1) into a single time slot problem (P4).

Lyapunov optimization involves minimizing the expected drift-plus-penalty value in each time slot. As defined in reference [9], let $L(t)$ be the sum of squares of backlog in all queues on slot t, which is called a *Lyapunov function*. Define $\Delta(L(t)) = E[\frac{1}{2}(Q_1(t+1)^2 - Q_1(t)^2)|Q_1(t)]$, being the difference in the Lyapunov function from one slot to the next, which is called a *Lyapunov drift*. Define the penalty term $E[U(t)|Q_1(t)]$. The drift-plus-penalty function is given by:

$$\Delta(L(t)) - VE[U(t)|Q_1(t)], \tag{20}$$

where $V \ge 0$ is a parameter that represents the importance of QoE. By solving the single time-slot Problem 2(P2) across the timeline from 1 to T_1:

$$\begin{aligned} \text{(P2)} \quad \arg\min_{\{r_{t,i}|i\in\{1,\dots,I\}\}} \quad & Q_1(t)(a_1(t) - b_1(t)) - VU(t) \\ \text{subject to(6)}, & \end{aligned} \tag{21}$$

we can minimize a bound on the drift-plus-penalty function, resulting in the following theoretical performance guarantee.

Theorem 1: The long-term video quality and queue length of virtual queue 1 satisfied:

$$\begin{aligned} \overline{U(t)} &\triangleq \lim_{T_1->\infty} \frac{1}{T_1} \sum_{t=1}^{T_1} E[U(t)] \\ &\ge U_1^*(\epsilon_1) - \frac{C+D}{V} \end{aligned} \tag{22}$$

$$\begin{aligned} \overline{Q_1(t)} &\triangleq \lim_{T_1->\infty} \frac{1}{T_1} \sum_{t=1}^{T_1} E[B(t) - S]] \\ &\le \frac{C+D}{\epsilon_1} + \frac{V(\overline{U(t)} - U_1^*(\epsilon_1))}{\epsilon_1}, \end{aligned} \tag{23}$$

where D represents the sum of the second-order moment bounds for the buffer's arrival and service rates.

[1]Traditional queue dynamic equation of the virtual queue are $Q(t) = max\{Q(t) - b(t), 0\} + a(t)$

Proof: see the appendix.

Similarly, the theorem holds for virtual queue 2. By solving the single time-slot Problem 3(P3) across the timeline from 1 to T_2:

$$\begin{aligned}(\text{P3}) \quad & \min_{\{r_{t,i}|i\in\{1,\ldots,I\}\}} \quad Q_2(t)(a_2(t)-b_2(t)) - VU(t) \\ & \text{subject to} \quad (6)\end{aligned} \tag{24}$$

The performance guarantee is:

$$\begin{aligned}\overline{U(t)} &\triangleq \lim_{T_2->\infty}\frac{1}{T_2}\sum_{t=1}^{T_2}E[U(t)] \\ &\geq U_2^*(\epsilon_2) - \frac{C+D}{V}\end{aligned} \tag{25}$$

$$\begin{aligned}\overline{Q_2(t)} &\triangleq \lim_{T_2->\infty}\frac{1}{T_2}\sum_{t=1}^{T_2}E[S-B(t)] \\ &\leq \frac{C+D}{\epsilon_2} + \frac{V(\overline{U(t)} - U_2^*(\epsilon_2)}{\epsilon_2}\end{aligned} \tag{26}$$

Lemma 2:

$$\lim_{T->\infty}\frac{T_1}{T_2} = \frac{\epsilon_2}{\epsilon_1} \tag{27}$$

Proof: see the appendix.

By integrating P2 from the timeline 1 to T_1 with P3 from the timeline 1 to T_2, we derive Problem 4(P4) from the timeline from 1 to T:

$$\begin{aligned}(\text{P4}) \quad & \arg\min_{\{r_{t,i}|i\in\{1,\ldots,I\}\}} \quad (B(t)-S)(a(t)-b(t)) - VU(t) \\ & \text{subject to(6)}\end{aligned} \tag{28}$$

By applying Theorem 1 and Lemma 2, solving P4 adheres to the following performance guarantee:

$$\begin{aligned}&\overline{U(t)} \triangleq \lim_{T->\infty}\frac{1}{T}\sum_{t=1}^{T}E[U(t)] \\ &\geq \lim_{T->\infty}\frac{1}{T}(\sum_{t=1}^{T_1}E[U(t)] + \sum_{t=1}^{T_2}E[U(t])\\ &\geq \frac{\epsilon_2}{\epsilon_1+\epsilon_2}U_1^*(\epsilon_1) + \frac{\epsilon_1}{\epsilon_1+\epsilon_2}U_2^*(\epsilon_2) - \frac{C+D}{V}\end{aligned} \tag{29}$$

$$\begin{aligned}\overline{B(t)} &\triangleq \lim_{T->\infty}\frac{1}{T}\sum_{t=1}^{T}E[B(t)] \\ &\leq S + \frac{C+D}{\epsilon_1} + \frac{V(\overline{U(t)} - U_1^*(\epsilon_1))}{\epsilon_1}\end{aligned} \tag{30}$$

$$\overline{B(t)} \geq S - \frac{C+D}{\epsilon_2} - \frac{V(\overline{U(t)} - U_2^*(\epsilon_2))}{\epsilon_2} \tag{31}$$

The performance bounds (29), (30) and (31) demonstrate an $[O(1/V), O(V)]$ QoE-buffer tradeoff: The larger the value of V, the higher the QoE. However, a larger V also implies a larger buffer boundary, increasing the probability of rebuffering and buffer overflow. S affects the upper and lower boundaries of the buffer. A larger S reduces the probability of rebuffering but increases the risk of buffer overflow.

Algorithm 1 Optimal tile rate determination of LSPO

Input:

Variables: Buffer occupancy $B(t)$; Tile weight $W_t = \{w_{t,i}|i\in\{1,...,I\}\}$

Parameters: Control parameter V; Stable Point S

Output: The determination of the best tile rate for the next chunk $R_t^{opt} = \{r_{t,i}|i\in\{1,...,I\}\}$

while the current chunk has not finished playing **do**
  //If the change is negative, $i=-1$.
  $i \leftarrow \underset{\{r_{t,i}+1|\{i\in\{1,\ldots,I\}\}}{argmax} \frac{\Delta((B(t)-S)(\frac{\sum_{i=1}^{I} r_{t,i}}{\sum_{i=1}^{I} v_1})+VU(t))}{change\_bit_i}$
  //'+1' refers to an increase of 1 in the quality level.
  if $i == -1$ **then**
    break
  end if
  $r_{t,i} \leftarrow r_{t,i}+1$
end while
return R_t^{opt}

D. Online Adaptation Algorithm

Problem 4 can be reformulated as a 0-1 knapsack problem, recognized as NP-hard. To facilitate rapid resolution, we solve this problem employing a heuristic algorithm, designated as Algorithm 1. This algorithm entails selecting the tile that maximizes the optimization objective for each increase in bitrate.

VI. Evaluation

In this section, we evaluate our online adaptation algorithm against other state-of-the-art algorithms on open-source datasets.

A. Experimental Setup

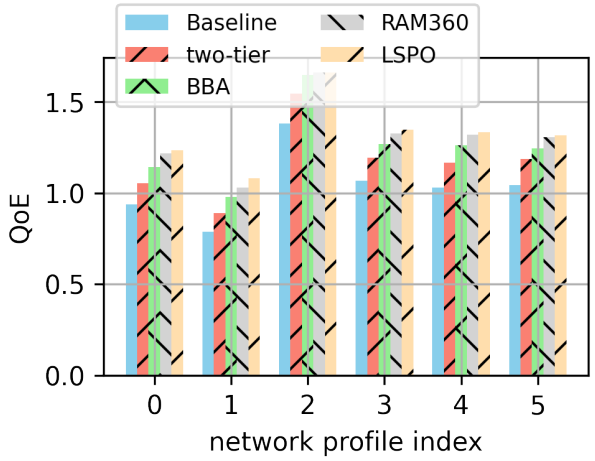

Fig. 6. Comparison of QoE across different algorithms under static mobility pattern

360-degree video Preparation. We select 360-degree videos from an open-source dataset [25]. The video bitrates

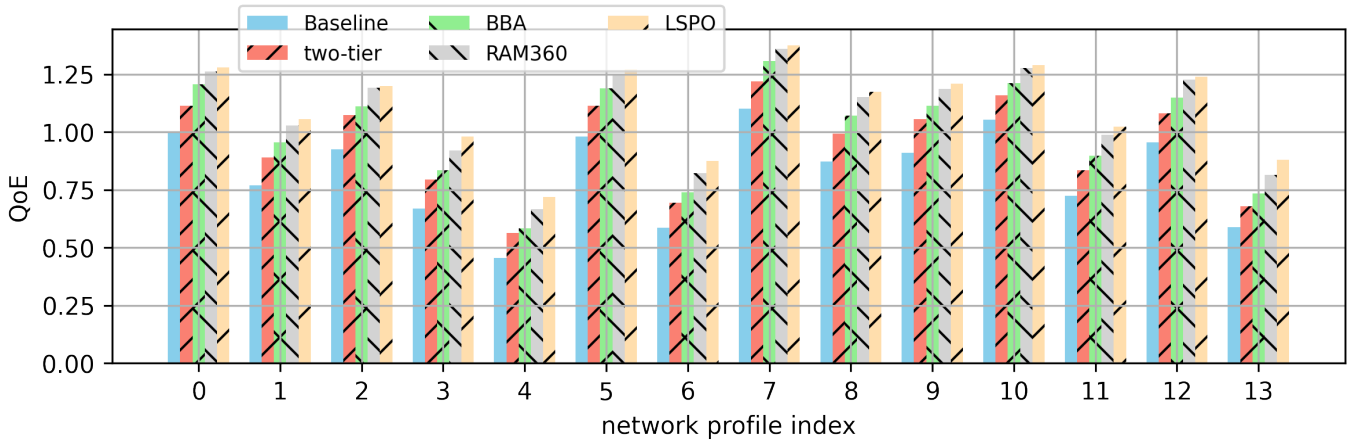

Fig. 7. Comparison of QoE across different algorithms under car mobility pattern

are considered to be encoded at [5, 10, 15, 20, 25, 30, 35, 40, 45]Mbps, with the lowest corresponding to 720p video bitrate and the highest to 4k video bitrate [26]. We adopt the conventional 8 x 8 tile configuration($N = 64$) as suggested in [24]. Each chunk has a duration of 1 second.

Viewport Prediction. Given the emphasis of this work on the design of bitrate adaptation algorithms, we utilize the PARIMA viewport prediction algorithm open-sourced in [10]. This algorithm leverages past user viewport and the trajectories of prime objects within the video content to predict future viewport.

Parameter Setting. The weight μ for viewport quality oscillation in (3) is set to 0.2. The maximum buffer size for the base layer is set to 30 seconds. The values of V and S are both set to 22.5 in (28) to achieve an optimal balance between QoE and buffer constrained.

Comparing Algorithms:

- **Baseline** [6]: When the base layer buffer falls below the threshold, all bandwidth is allocated to download the base layer. When the base layer buffer exceeds the threshold, the bandwidth is allocated to download the base layer and the enhancement layer of the next chunk.
- **BBA** [8]: A linear mapping function is utilized to map the base layer buffer length to the corresponding decision's video bitrate.
- **2Tier** [20]: First, the PID control algorithm manages buffer and derives the available bandwidth. Then, based on the accuracy of viewport prediction, the proportion of bandwidth allocated between the base layer and the enhancement layer is determined.
- **RAM360** [24]: It adopts the Lyapunov Optimal. Its optimization objective is the linear weighted sum of video quality and video rebuffering.

B. Performance under real-world bandwidth traces

We evaluate the performance of LSPO under real-world bandwidth dataset [27] and user trajectories [25]. We utilize the downstream bandwidth data provided by Netflix for video streaming under 5G conditons as our bandwidth dataset during playback. The dataset is generated from two mobility patterns: static and car.

QoE calculate according to (3). As shown in Fig. 6, LSPO outperforms the other state-of-the-art algorithms in static mobility pattern. We tested video playback over a duration of 1000 seconds, during which, under the majority of network profile, playback is smooth without any interruptions. Only network profile 1 experience 17 instances of rebuffering; nonetheless, all tested algorithms faced these 17 rebuffering. This is attributed to the network's continuous bandwidth falling short of the minimum bitrate requirements, consequently leading to buffer exhaustion.

To validate the necessity of a long buffer, we conducted experiments with a buffer length of 3 seconds [4]. Out of 1000 video chunk downloads, more than 100 instances experience rebuffering. This is attributed to the probability of bandwidth consecutively falling below the minimum bitrate requirement decreasing as the number of occurrences increases, hence a larger buffer size can effectively prevent video rebuffering.

As shown in Fig. 7, LSPO also outperforms the other state-of-the-art algorithms in car mobility pattern. Only network profile 6 and 13 respectively experience 5 and 1 instances of rebuffering. Similarly, all tested algorithms experience these rebuffering. This result strengthen our effectiveness.

VII. Conclusion

The paper concludes that LSPO represents a significant advancement in bitrate adaptation for 360° video streaming, addressing the limitations of existing approaches by ensuring high QoE and efficient buffer management. Future work can explore refining the parameters of LSPO under various network scenarios.

APPENDIX

A. Proof of Lemma 1

$Q(t)$ follows to the new queue dynamic equation:

$$Q(t+1) = [Q(t) - b(t) < 0 ? \beta(t) : Q(t) - b(t)] + a(t) \quad (32)$$

The actual service rate $\tilde{b}(t)$ of $Q(t)$ is:

$$\tilde{b}(t) \triangleq [(Q(t) - b(t) < 0)?Q(t) - \beta(t) : b(t)] \leq b(t) \tag{33}$$

Therefore, rewriting equation (32)without the non-linear operator as follow:

$$Q(t+1) = Q(t) - \tilde{b}(t) + a(t) \tag{34}$$

If $\lim_{t->\infty} \frac{Q(t)}{t} = 0$, then:

$$\begin{aligned} \frac{1}{t}Q(t) - \frac{1}{t}Q(1) &= \frac{1}{t}\sum_{\tau=1}^{t} a(\tau) - \frac{1}{t}\sum_{\tau=1}^{t} \widetilde{b}(\tau) \\ &\geq \frac{1}{t}\sum_{\tau=1}^{t} a(\tau) - \frac{1}{t}\sum_{\tau=1}^{t} b(\tau) \end{aligned} \tag{35}$$

Thus, it follows that $E[a(t)] \leq E[b(t)]$.

If $E[a(t)] \leq E[b(t)]$: Let's assume that $Q_\epsilon(t)$ follows:

$$Q_\epsilon(t+1) = [Q_\epsilon(t) - b(t) < 0?\beta(t) : Q_\epsilon(t) - b(t)] + a_\epsilon(t) \tag{36}$$

$$a_\epsilon(t) = a(t) + (E(b(t)) - E(a(t))) + \epsilon(\epsilon > 0) \tag{37}$$

where ϵ is any positive real number. The arrival rate of $Q_\epsilon(t)$ is:

$$E(a_\epsilon(\tau)) = E(b(t)) + \epsilon > E(b(t)) \tag{38}$$

Lemma 3:

$$\begin{aligned} &E[a(t)] \geq E[b(t)] \\ \Rightarrow &\lim_{t->\infty} \frac{Q(t)}{t} = E[a(t)] - E[b(t)] \end{aligned} \tag{39}$$

The full proof can be found in [9].

Therefore, it follows that:

$$\lim_{t->\infty} \frac{Q_\epsilon(t)}{t} = E(a_\epsilon(t)) - E(b(t)) = \epsilon \tag{40}$$

Since

$$Q(t) \leq Q_\epsilon(t) + C \tag{41}$$

(can be proven by mathematical induction.) and ϵ is any positive real number, it follows that

$$\lim_{t->\infty} \frac{Q(t)}{t} \leq \frac{Q_\epsilon(t)}{t} + \frac{C}{t} = \epsilon \tag{42}$$

Proof completed.

B. Proof of Theorem 1

Since the first and second moments of the bandwidth are confined within certain limits:

$$E[\frac{a_1(t)^2 + b_1(t)^2}{2}] = E[\frac{[a(t) - b(t)]^2}{2}] \leq D \tag{43}$$

$$E[\frac{\beta_1(t)^2}{2}] \leq C \tag{44}$$

The following inequalities hold:

$$\begin{aligned} &L(t+1) = \frac{1}{2}[Q_1(t+1)]^2 \\ &= \frac{1}{2}[Q_1(t) - b_1(t) < 0?\beta_1(t) : Q_1(t) - b_1(t)]^2 \\ &\leq \frac{1}{2}\{Q_1(t)^2 - 2Q_1(t)b_1(t) + b_1(t)^2 + \beta_1(t)^2\} \end{aligned} \tag{45}$$

Since we take an action that minimizes a bound on the drift-plush-penalty function, the decision at each time slot follows Equation (21). As a result, our decision is less than or equal to any other decision, including the optimal P-only algorithm, denoted as $\alpha_1^*(t)$. According to equations (45):

$$\begin{aligned} &\Delta(L(t)) - VE[U(t)|Q_1(t)] \\ \leq &E[\frac{b_1(t)^2 + \beta_1(t)^2}{2}|Q_1(t)] - E[Q_1(t)b_1(t)|Q_1(t)] \\ &- VE[U(t)|Q_1(t)] \\ \leq &C + D - E[Q_1(t)b_1(\alpha_1^*(t))|Q_1(t)] \\ &- VE[U(\alpha_1^*(t))|Q_1(t)] \end{aligned} \tag{46}$$

According to equation (19) and applying the law of iterated expectations:

$$\begin{aligned} &E[L(t+1) - L(t)] - VE[U(t)] \\ &\leq C + D - E[Q_1(t)]\epsilon_1 - VE[U_1^*(\epsilon_1)] \end{aligned} \tag{47}$$

Summing over $1, ..T_1$:

$$\begin{aligned} &E[L(T_1+1) - L(1)] - V\sum_{t=1}^{T_1} E[U(t)] \\ &\leq (C+D)T_1 - \sum_{t=1}^{T_1} E[Q_1(t)]\epsilon_1 - VT_1E[U_1^*(\epsilon_1)] \end{aligned} \tag{48}$$

Therefore:

$$\frac{1}{T_1}\sum_{t=1}^{T_1} E[U(t)] \geq E[U_1^*(\epsilon_1)] - \frac{C+D}{V} - \frac{E[L(1)]}{VT_1} \tag{49}$$

$$\begin{aligned} &\frac{1}{T_1}\sum_{t=1}^{T_1} E[Q_1(t)] \\ &\leq \frac{C + D + V(\frac{1}{T_1}\sum_{t=1}^{T_1} E[U(t)] - U_1^*(\epsilon_1))}{\epsilon_1} \\ &+ \frac{E[L(1)]}{\epsilon_1 T_1} \end{aligned} \tag{50}$$

In conclusion, we obtain equations (23) and (24).

Proof completed.

C. Proof of Lemma 2

Equations (23) and (26) imply a bidirectional constraint in the buffer, which can be formulated as:

$$\mu_1 \leq \lim_{T->\infty} \frac{1}{T}\sum_{t=1}^{T} B(t) \leq \mu_2 \tag{51}$$

$$\begin{aligned} \lim_{T->\infty} \frac{1}{T}\sum_{t=1}^{T} B(t) &= \frac{1}{T}\sum_{t=1}^{T}\sum_{\tau=1}^{t}(a(\tau) - b(\tau)) \\ &= \frac{T+1}{2}[E(a(t)) - E(b(t))] \end{aligned} \tag{52}$$

Hence, it follows that:

$$\begin{aligned} &E(a(t)) - E(b(t)) = \frac{1}{T}\sum_{t=1}^{T}(a(t) - b(t)) \\ &= \frac{1}{T}\{\sum_{t=1}^{T_1}(a_1(t) - b_1(t)) - \sum_{t=1}^{T_2}(a_2(t) - b_2(t))\} \end{aligned} \tag{53}$$

When $T->\infty$, from equation (19), it follows that:

$$\begin{aligned}\frac{1}{T_1}\sum_{t=1}^{T_1}[a_1(t)-b_1(t)] &= -\epsilon_1 \\ \frac{1}{T_2}\sum_{t=1}^{T_2}[a_2(t)-b_2(t)] &= -\epsilon_2\end{aligned} \tag{54}$$

Therefore:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T_1}(a_1(t)-b_1(t)) &= \frac{1}{T}\sum_{t=1}^{T_2}(a_2(t)-b_2(t)) \\ \frac{T_1}{T}\epsilon_1 &= \frac{T_2}{T}\epsilon_2 \\ \frac{T_1}{T_2} &= \frac{\epsilon_2}{\epsilon_1}\end{aligned} \tag{55}$$

Proof completed.

REFERENCES

[1] G. V. Research. (2023) Virtual reality market size, share & analysis report, 2023-2030. Accessed: 2023-09-04. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market

[2] J. Yi, S. Luo, and Z. Yan, "A measurement study of youtube 360 live video streaming," in *Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, Amherst, MA, USA*, 2019, pp. 49–54.

[3] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India*, 2018, pp. 99–114.

[4] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, "Pano: Optimizing 360 video streaming with a better understanding of quality perception," in *Proceedings of the ACM Special Interest Group on Data Communication, Beijing, China*, 2019, pp. 394–407.

[5] M. Almquist, V. Almquist, V. Krishnamoorthi, N. Carlsson, and D. Eager, "The prefetch aggressiveness tradeoff in 360 video streaming," in *Proceedings of the 9th ACM Multimedia Systems Conference, Amsterdam, The Netherlands*, 2018, pp. 258–269.

[6] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using scalable video coding," in *Proceedings of the 2017 ACM on Multimedia Conference, Mountain View, CA, USA*, 2017, pp. 1689–1697.

[7] P. K. Yadav, A. Shafiei, and W. T. Ooi, "Quetra: A queuing theory approach to dash rate adaptation," in *Proceedings of the 2017 ACM on Multimedia Conference, Mountain View, CA, USA*, 2017, pp. 1130–1138.

[8] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *ACM SIGCOMM 2014 Conference, Chicago, IL, USA*, 2014, pp. 187–198.

[9] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Springer Nature, 2022.

[10] L. Chopra, S. Chakraborty, A. Mondal, and S. Chakraborty, "Parima: Viewport adaptive 360-degree video streaming," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia*, 2021, pp. 2379–2391.

[11] A. Nguyen, Z. Yan, and K. Nahrstedt, "Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction," in *2018 ACM Multimedia Conference on Multimedia Conference, Seoul, Republic of Korea*, 2018, pp. 1190–1198.

[12] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 6th Workshop on All Things Cellular - Operations, Applications and Challenges, New York City, New York, USA*, 2016, pp. 1–6.

[13] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive view generation to enable mobile 360-degree and vr experiences," in *Proceedings of the 2018 Morning Workshop on Virtual Reality and Augmented Reality Network, Budapest, Hungary*, 2018, pp. 20–26.

[14] Y. Zhang, Y. Guan, K. Bian, Y. Liu, H. Tuo, L. Song, and X. Li, "Epass360: Qoe-aware 360-degree video streaming over mobile devices," *IEEE Transactions on Mobile Computing*, vol. 20, no. 7, pp. 2338–2353, 2020.

[15] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive adaptive streaming to enable mobile 360-degree and vr experiences," *IEEE Transactions on Multimedia*, vol. 23, pp. 716–731, 2020.

[16] Y. Zhang, P. Zhao, K. Bian, Y. Liu, L. Song, and X. Li, "Drl360: 360-degree video streaming with deep reinforcement learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France,*. IEEE, 2019, pp. 1252–1260.

[17] M. Tang and V. W. Wong, "Online bitrate selection for viewport adaptive 360-degree video streaming," *IEEE Transactions on Mobile Computing*, vol. 21, no. 7, pp. 2506–2517, 2020.

[18] F. Duanmu, E. Kurdoglu, S. A. Hosseini, Y. Liu, and Y. Wang, "Prioritized buffer control in two-tier 360 video streaming," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, Los Angeles, CA, USA*, 2017, pp. 13–18.

[19] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, "Multi-path multi-tier 360-degree video streaming in 5g networks," in *Proceedings of the 9th ACM Multimedia Systems Conference, Amsterdam, The Netherlands*, 2018, pp. 162–173.

[20] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and Dai, "A two-tier system for on-demand streaming of 360 degree video over dynamic networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 43–57, 2019.

[21] Z. Jiang, X. Zhang, W. Huang, H. Chen, Y. Xu, J.-N. Hwang, Z. Ma, and J. Sun, "A hierarchical buffer management approach to rate adaptation for 360-degree video streaming," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2157–2170, 2019.

[22] Z. Jiang, X. Zhang, Y. Xu, Z. Ma, J. Sun, and Y. Zhang, "Reinforcement learning based rate adaptation for 360-degree video streaming," *IEEE Transactions on Broadcasting*, vol. 67, no. 2, pp. 409–423, 2020.

[23] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "Bola: Near-optimal bitrate adaptation for online videos," *IEEE/ACM transactions on networking*, vol. 28, no. 4, pp. 1698–1711, 2020.

[24] H. Zhang, Y. Ban, Z. Guo, K. Chen, and X. Zhang, "Ram360: Robust adaptive multi-layer 360 video streaming with lyapunov optimization," *IEEE Transactions on Multimedia*, vol. 25, pp. 4225–4239, 2023.

[25] 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, Taipei, Taiwan*, 2017, pp. 193–198.

[26] L. Zhang, H. Guo, Y. Dong, F. Wang, L. Cui, and V. C. Leung, "Collaborative streaming and super resolution adaptation for mobile immersive videos," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications, New York City, NY, USA*. IEEE, 2023, pp. 1–10.

[27] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: a 5g dataset with channel and context metrics," in *Proceedings of the 11th ACM Multimedia Systems Conference, Istanbul, Turkey*, 2020, pp. 303–308.