A Two-Tier System for On-Demand Streaming of 360 Degree Video Over Dynamic Networks

Liyang Sun, Fanyi Duanmu, Yong Liu, *Fellow, IEEE*, Yao Wang, *Fellow, IEEE*, Yinghua Ye, *Senior Member, IEEE*, Hang Shi, and David Dai

***Abstract*— 360° video on-demand streaming is a key component of the emerging virtual reality and augmented reality applications. In such applications, sending the entire 360° video demands extremely high network bandwidth that may not be affordable by today's networks. On the other hand, sending only the predicted user's field of view (FoV) is not viable as it is hard to achieve perfect FoV prediction in on-demand streaming, where it is better to prefetch the video multiple seconds ahead, to absorb the network bandwidth fluctuation. This paper proposes a two-tier solution, where the base tier delivers the entire 360° span at a lower quality with a long prefetching buffer, and the enhancement tier delivers the predicted FoV at a higher quality using a short buffer. The base tier provides robustness to both network bandwidth variations and FoV prediction errors. The enhancement tier improves the video quality if it is delivered in time and FoV prediction is accurate. We study the optimal rate allocation between the two tiers and buffer provisioning for the enhancement tier to achieve the optimal trade-off between video quality and streaming robustness. We also design periodic and adaptive optimization frameworks to adapt to the bandwidth variations and FoV prediction errors in realtime. Through simulations driven by real LTE and WiGig network bandwidth traces and user FoV traces, we demonstrate that the proposed two-tier systems can achieve a high-level of quality-of-experience in the face of network bandwidth and user FoV dynamics.**

***Index Terms*— 360° video, on-demand video streaming, virtual reality.**

I. Introduction

VIRTUAL Reality (VR) and Augmented Reality (AR) technologies have become popular in recent years. Many VR/AR applications are rapidly commercialized in different sectors, including movie and gaming, education and training, healthcare, advertising and social media, etc. Many VR/AR applications involve on-demand streaming of 360° video. Therefore, the delivery of ultra high quality 360° video is critically important for the wide adoption of VR/AR. Compared with traditional video streaming, 360° video streaming confronts unique new challenges. Firstly, to deliver an immersive VR experience, 360° video has much higher bandwidth requirement. A premium quality 360° video with 120 frames-per-second and 24K resolution can easily consume a bandwidth of multiple Gigabits-per-second (Gbps) [1]. For smooth rendering, video has to be streamed consistently at high rate. Meanwhile, 360° video streaming is constantly driven by user Field-of-View (FoV) changes: at any given time a user only watches a video scene within a FoV centered at certain direction and with limited horizontal and vertical spans; a user can change her FoV at any time, and expect to see the video scene in the new FoV immediately after her head movement. Recent subjective user study has suggested that if the video rendering latency after a FoV change, the so-called *Motion-to-Photon (MTP) latency*, is above twenty milliseconds, users will experience motion sickness [1]. This imposes stringent latency requirement for 360° video delivery.

In this paper, we propose a novel two-tier 360° video streaming framework to maximize the rendered video quality, while maintaining the streaming continuity and robustness against the inherent dynamics in both user FoV and network bandwidth. In the proposed framework, the server codes and stores video segments in two tiers: the base tier (BT) contains video chunks covering the full 360° span coded with a low rate, whereas the enhancement tier (ET) includes video chunks covering overlapping viewports with limited view spans and are coded with multiple rates. The receiver will download the BT chunks with a long prefetching buffer to combat bandwidth variations, and request the ET chunks that cover the predicted FoVs using a short prefetching buffer to ensure sufficiently high view prediction accuracy. At the display time for each video segment, the user's FoV will be rendered in high quality from the ET chunk if it is already in the buffer and the FoV prediction is correct. Otherwise, the BT chunk will be used to generate a low quality rendering. The base tier stream provides robustness to both network bandwidth fluctuation and FoV prediction errors. Note that this robustness is achieved with a slight redundancy, as the predicted FoV region is delivered twice: first in the base tier and again in the enhancement tier. This redundancy can be minimized by using layered coding between the BT and ET chunks. However, for system operation simplicity, non-layered coding may be preferred in practice.

Within the proposed two-tier streaming architecture, we have investigated two-tier rate allocation and chunk scheduling to achieve the optimal trade-off between video quality and streaming robustness. Our main contributions are as follows.

1) We propose a novel two-tier on-demand 360° video streaming framework which features prioritized BT chunk downloading and opportunistic ET chunk downloading to provide robustness to both network and

Manuscript received August 15, 2018; revised December 21, 2018; accepted January 29, 2019. Date of publication February 12, 2019; date of current version March 11, 2019. This work was supported in part by the USA NSF under Award CNS-1816500. This paper was recommended by Guest Editor J. Boyce. *(Corresponding author: Liyang Sun.)*

L. Sun, F. Duanmu, Y. Liu, and Y. Wang are with the Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: ls3817@nyu.edu; fanyi.duanmu@nyu.edu; yongliu@nyu.edu; yw523@nyu.edu).

Y. Ye, H. Shi, and D. Dai are with Huawei Technologies, Santa Clara, CA 95050 USA (e-mail: yinghua.ye@huawei.com; hang.shi@huawei.com; david.h.dai@huawei.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2019.2898877

FoV dynamics, while maximizing the rendered video quality.
2) We analytically study the optimization of the target ET buffer length and rate allocation between the BT and ET. Our study brings forth important understanding about the interplay between the key components of 360° video streaming, including FoV prediction accuracy, chunk delivery rate, rate allocation, and the ET buffer length (which equals to FoV prediction horizon).
3) We develop algorithms that dynamically adjust the rate allocation and ET buffer length based on the real-time measurement of the network bandwidth statistics and FoV prediction accuracy as a function of the ET buffer length.
4) We conduct extensive experiments driven by real WiGig 802.11ad and LTE bandwidth traces with different levels of volatilities and real user FoV traces with diverse head movement patterns, and demonstrate that the proposed system provides substantial improvement in the QoE over two benchmark systems (streaming the entire 360° video and streaming only the predicted FoV).

Preliminary results for the proposed system have been described in [2]–[4]. Specifically, the two-tier system was first described in [2] without optimization of the chunk scheduling and other system parameters; the prioritized chunk scheduling algorithm was described in [3]; and optimization of buffer length and rate allocation was described in [4]. This paper brings these prior results together, and also for the first time presents dynamic optimization of the buffer length and rate allocation based on real-time measurement of the network and user behaviors, which is critical for the adoption of the proposed system in a practical dynamic network.

The rest of the paper is organized as follows. Background and related work on 360° video streaming are reviewed in Sec. II. The optimization of the buffer length and rate allocation is studied in Sec. III. Dynamic adaptation is presented in Sec. III-D. Experimental results are presented in Sec. V. The paper is concluded in Sec. VI.

II. Background and Related Work

In recent years, numerous solutions have been proposed to address 360° video compression and delivery, as categorized into the following two major categories:

A. Source Representation

In a typical 360° video compression and delivery framework, the input 360° videos, represented in a native projection format, e.g., equirectangular (ERP), are sometimes converted into another projection format, e.g., cubemap (CMP) [5], octahedron (OHP) [6], etc. and frame-packed before being fed into existing video codecs. The intermediate projection format is important and would potentially improve the representation efficiency and coding performance. For example, Facebook proposed to use the cube-map [5] and pyramid [7] projection methods and encoding schemes in 2016, to specifically address the on-demand 360° video streaming, with 25% and 80% reported compression improvements, respectively. The Joint Video Exploration Team (JVET) and others also proposed a few projection solutions for next-generation video coding for 360° video, including Icosahedral projection (ISP) [8], Segmented Sphere Projection (SSP) [9], Truncated Square Pyramid Projection (TSP) [10], Octahedron Projection (OHP) [6], Hybrid Cubemap Projection [11], [12], etc.

B. Viewport-Adaptive Streaming

Facebook proposed a FoV-adaptive encoding and streaming framework [7] using a pyramid projection solution, in which the base of the pyramid is the full-resolution FoV and the sides of the pyramid represent non-FoV region in gradually lower resolutions. The top of the pyramid corresponds to the point directly opposite from the center of the predicted FoV. In their system, 30 pyramid videos covering different viewports are generated and each is encoded at 5 different rates to accommodate different viewing directions and different network conditions. Each viewport video in the pyramid representation is unwrapped and frame-packed into a rectangular format to feed into video encoder. An 80% file size reduction is reported compared with ERP representation. In [13], multi-resolution ERP and CMP videos are generated with different rate allocations for FoV and non-FoV regions to realize adaptive viewport streaming. Similar viewport adaptive 360° video streaming solutions can be found in [14]–[16], etc. In recent years, the tile-based solutions become popular and widely used for viewport adaptive streaming and provide flexible rate allocation and delivery prioritization. In tile-based solutions, usually the entire 360° video is divided into non-overlapping small rectangular regions (i.e., “tiles”), and coded independently at different bitrates. For example, in [17], a High Efficiency Video Coding (HEVC) compliant approach is proposed for efficient coding and streaming of stereoscopic VR contents, in which video pictures are partitioned into tiles and only the required tiles corresponding to the FoV regions are transmitted in high resolution, while the remaining tiles are transmitted in low resolution. In [18], several tile-based encoding solutions are proposed, including both scalable coding scheme and simulcast coding scheme. In [19], a view prediction based framework is proposed by only fetching the video portions desirable to the end user to reduce the bandwidth consumption. A dynamic video chunk adaptation scheme is implemented to adjust tile coverage based on the view prediction accuracy. An estimated 80% maximum rate reduction (compared with naive full-360° video delivery) is reported without considering the coding efficiency loss due to video tiling. Additional tile-based solutions can be found in [20]–[27], etc.

We would like to clarify that there is a fundamental difference between the proposed two-tier system and the above-mentioned viewport-based systems. Although these systems are designed to stream tiles/regions which are not under predicted FoV in lower quality to prevent “black" screens when the FoV prediction is wrong, these regions are requested together with the predicted FoV regions for the same video segment. To reach sufficiently high FoV prediction accuracy, these systems have to limit the prefetching to a very short time. Under very dynamic network conditions, the requested chunks may not arrive before the display deadline, leading to video freezing. Our two-tier system, by transmitting the BT and ET streams using different buffers, can ensure with a high probability that the user’s FoV can always be rendered smoothly, albeit sometimes at a lower quality. Besides, our system is source-representation-agnostic, and can work with any effective methods for coding the viewport videos at the ET and 360° videos at the BT. Our system can also utilize any efficient FoV prediction algorithms.

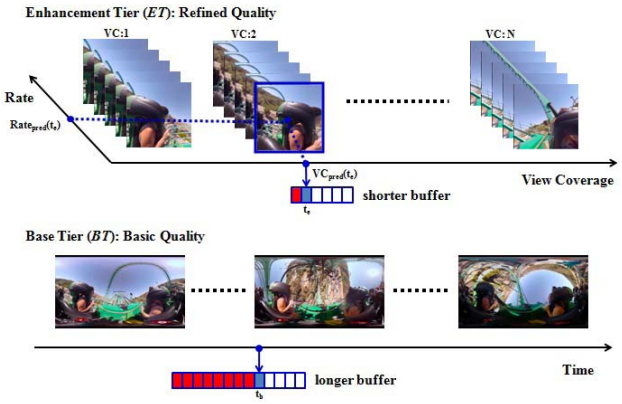


Fig. 1. Two-tier 360° video streaming framework.

III. Two-Tier Streaming Framework

A. System Overview

As illustrated in Fig. 1, in the proposed two-tier framework, a 360° video is partitioned into non-overlapping time segments with each segment encoded as a BT chunk and multiple ET chunks. A BT chunk encodes the entire 360° view span (360° × 180°) at a low bitrate to provide the basic quality. BT chunks for future time segments are pre-fetched into a long streaming buffer to cope with network bandwidth variations and guarantee that any desired FoV can be rendered with minimum stalls at the client. Each ET chunk encodes video within a viewport (VP) with a certain view coverage (VC) (e.g., 135° × 135°) centered at a certain direction. To provide quality differentiation and adaptation to varying network bandwidth, multiple ET chunks are generated for the same viewport, but coded at different bitrates. For complete coverage and smooth transition, the viewports of all ET chunks in the same time segment are *overlapping* and cover the whole 360° view span. An ET chunk can be used for rendering at the client side only if it covers (completely or partially) the user's actual FoV. Since it is difficult to predict a user's view direction far into the future, only ET chunks in the near future will be pre-fetched. All the pre-coded BT and ET chunks are stored in the streaming server. During the streaming, the client dynamically requests the precoded chunks from BT or ET to download, according to the predicted viewing direction, the predicted download bandwidth for the next request interval, and the buffer status of each tier. Our current system uses a fixed view coverage for the ET viewports. More generally, viewports with different view coverages can also be coded and stored in the server. Depending on the anticipated FoV prediction accuracy, the client can choose between ET chunks with different view coverages.

Compared with the traditional single-tier solution, in the proposed framework, the ET and BT video chunks are stored in two dedicate buffers. The longer BT buffer provides additional robustness against network variation to maximally avoid video freeze, whereas the shorter ET buffer guarantees a higher view prediction accuracy to minimize the tiles needed to be transmitted. At the client-side, the player decodes, synchronizes the BT and ET chunks (according to chunk offset) and combines the two tiers (according to the pre-defined viewport direction and viewport span) for the final rendering and display. Though there is a slight system complexity introduced, for example, during parallel tier decoding and cross-tier chunk synchronization, however, as presented later in Sec.V-E, the achieved QoE improvement can significantly justify and outweigh such complexity overhead.

Under the two-tier framework, multiple challenging problems need to be addressed, including video coding (how to partition the 360° span into overlapping viewports and how to code BT and ET chunks?), chunk scheduling (from which tier and at what rate to request the next chunk?), rate allocation (what rates to use for coding the BT and ET chunks?), buffer setting (what target buffer length to use for the BT and ET streams?), and FoV prediction. Our solutions to these problems are presented in the following subsections.

B. Video Coding

We assume that a 360° video is represented in the equirectangular projection (ERP) format, although similar approaches can be derived for other projection formats. To code a 360° video segment in the BT, we will code the entire ERP plane as a 2D video. For coding the viewports in the ET, there are multiple choices, including tile-based vs. viewport-based, layered vs. non-layered coding (also known as simulcast). We discuss the pros and cons of these options in this section. The proposed system can work with any one of these options. However, tile-based, non-layered coding is used in our experimental studies reported in Sec. V. In Appendix, we describe the operational quality-rate points obtained from our coding experiments and show that they can be approximated well by logarithmic quality-rate models, which are assumed in our derivation of the optimal rate allocation solution in Sec. IV-A.

1) Tile vs. Viewport Based Coding for the Enhancement Tier: In tile-based coding, we divide an entire ERP frame into multiple non-overlapping tiles, each covering a small rectangular region on the ERP. Each tile is coded independent of the others. At the enhancement tier, we will send all the tiles necessary to cover a desired viewport. Note that generally, the number of tiles needed to cover different viewports differs depending on the viewport direction. One benefit of tile-based coding is that one can easily construct viewports of increasing view spans by adding additional tiles. However, because spatial and temporal prediction must be limited within a single tile, the coding efficiency is compromised (requiring more transmission bandwidth to achieve a similar quality).

In viewport-based coding, each viewport is coded in its entirety to allow spatial and temporal prediction over a larger spatial span, so as to maximize the coding efficiency and consequently reduce bandwidth consumption for transmitting a requested viewport. However, it is important to note that the region corresponding to an arbitrary viewport is not a rectangular region on the ERP plane. One option is to render the 2D view corresponding to a viewport, and code the rendered video. However, if the actual viewing direction of a user is not the same as the viewport center, the delivered 2D view needs to be projected back to the ERP plane and merge with the decoded ERP from the BT data, and this merged ERP will then be used to render the desired 2D view. This extra projection can add unnecessary distortion and also incur additional complexity. Another option is to find the minimal rectangular region in the ERP that covers the desired viewport and set all unneeded pixels to zero, and then code the zero-padded rectangular region. Such an approach may lead to coding overhead similar to tile-based coding and yet is more complex. Because of these issues, we have chosen to use tile-based coding in the experimental study reported in this paper.

2) Layered Coding vs. Simulcast Across Tiers: With simulcast, the video in each tier is coded independently. When a user receives an ET chunk and the chunk covers the user's FoV completely for this video segment, the receiver only needs to decode this ET chunk to render the desired FoV. In this case, the corresponding BT chunk is not used. Otherwise, if either the ET chunk does not arrive in time or it does not overlap with the user's FoV, the receiver can decode the BT chunk for the same video segment. If the ET chunk is available but only partially covers the user's FoV, the receiver needs to decode the BT and ET chunks, each to the ERP plane, and merge them (with the ET decoded pixels overwriting the BT decoded pixels). With layered coding, an ET chunk will be coded relative to the corresponding portion in the decoded ERP video from the BT chunk. With this approach, if the arrived ET chunk covers the user's FoV either completely or partially, the receiver must decode both the BT and ET chunks to render the video at the desired FoV. Therefore, both encoding and decoding is more complex with layered coding.

At the outset, one may think that layered coding will improve network bandwidth utilization, as BT bits are never wasted, and ET bits are used to improve the quality provided by the BT bits. However, in practice, layered coding incurs coding overhead (i.e. more bits are needed to reach the same quality). Denote the normalized rates (bits/second/degree) for the BT chunk and ET chunk respectively as $\tilde{R}_b$ and $\tilde{R}_{e,SC}$, with simulcast. If we use layered coding, to reach the same quality at the ET, the total rate $\tilde{R}_b + \tilde{R}_{e,LC}$ can be expressed as $(1+\delta)\tilde{R}_{e,SC}$, where δ represents the bitrate overhead of layered coding compared to non-layered coding. The relative difference in the rate by the two systems is thus $(\tilde{R}_{e,SC} - \tilde{R}_{e,LC})/\tilde{R}_{e,SC} = \tilde{R}_b/\tilde{R}_{e,SC} - \delta$. Therefore the potential bandwidth savings by layered coding depends on the overhead δ and the ratio $\tilde{R}_b/\tilde{R}_{e,SC}$. Using the latest HEVC scalable coding extension model (SHVC) [28], δ has been reported to be about 14% for SNR scalability and higher for spatial scalability. Note that because the entire ERP is coded as a single unit in the BT, while a viewport in the ET is a subregion in the ERP, which can be coded either together or in separate tiles, one cannot perform layered coding directly in the bitstream domain as in the SNR or spatial scalability mode of SHVC. This will further increase the layered coding overhead. On the other hand, as shown in Sec. IV-A, the ratio $\tilde{R}_b/\tilde{R}_e$ under optimal rate allocation is typically quite small. From the experimental results shown in Table V and VI, on average, this ratio ranges between 7% to 14%. In general, this ratio can be close to (either larger or smaller than) the expected range of δ. Therefore, layered coding is not likely to bring bandwidth savings, in spite of the added complexity for encoding and decoding.

Another complication with layered coding is that ET encoding depends on the BT coding rate. Different sets of ET chunks need to be generated for each BT rate. Because the BT chunks are typically fetched much earlier than the ET chunks for the same video segment in the two-tier system, one cannot independently adapt the BT and ET rates, in response to the network bandwidth dynamics. On the other hand, with simulcast, because BT and ET chunks are coded independently, they can be decoded separately at the client, which will greatly simplify dynamic rate adaptation of the system, as will become clear in Sec. III-D. Because of these reasons, simulcast is adopted in our system design.

C. Prioritized Chunk Scheduling Algorithm

To guarantee the delivery of the BT chunks before their display deadlines, we propose the prioritized chunk scheduling strategy shown in Algorithm 1, which is motivated by the 2D streaming algorithm [29], and first described for two-tier streaming in [3]. Essentially, at time t when a new chunk needs to be requested, if the BT buffer length $B_b(t)$ is below the target buffer length B_b^T, the client will sequentially download BT chunks until the target length B_b^T is reached; otherwise, an ET chunk will be requested at the predicted direction P_e, and the rate of the ET chunk R_e is regulated by a P-I controller, driven by the estimated real-time bandwidth BW_t, the current ET buffer length $B_e(t)$ and target ET buffer length B_e^T. The rate of ET chunk to be downloaded at time t is:

$$u(t) = K_P(B_e(t) - B_e^T) + K_I \sum_{i=t-m}^{t} (B_e(i) - B_e^T), \quad (1)$$

$$\hat{R}(t) = \min\left[u(t)+1, \frac{\Delta_e}{\tau}\right] \cdot BW_t, \quad (2)$$

where K_P and K_I are the proportional and integration gain control factors, respectively, m is the integral interval and $u(t)$ is the control signal determined by both the current and historical ET buffer status. Δ_e is the remaining time till the display deadline of the ET chunk to be downloaded, and τ is the chunk duration. Eventually, the largest available ET rate that is equal or less than $\hat{R}(t)$ will be chosen. In addition, if both BT and ET buffer lengths are greater than their respective upperbound, the system will stay idle for some duration δ_t. δ_b or δ_e refers to the downloading time of BT or ET chunk. Experimental results in [3] and [4] demonstrate that this chunk scheduling algorithm can achieve good trade-off between the delivered video quality and the robustness against bandwidth variations and user view dynamics, even when the other system parameters (e.g. rate allocation and target buffer length) are not optimized.

Algorithm 1 Two-Tier 360° Video Streaming

```
Initialization at t = 0;
while (One chunk downloading is finished or t = 0) and display is not terminated do
    if B_b(t) <= B_b^T then
        Download next BT chunk C_b;
        t ← t + δ_b;
    else
        if B_e(t) <= B_e^(T) then
            Predict bandwidth BW_t
            Predict FoV P_e for next ET chunk C_e;
            Choose the rate version following Eq. (1) and (2) based on B_e^T, B_e(t) and BW_t;
            Request for next ET chunk C_e;
            t ← t + δ_e
        else
            t ← t + δ_t;
        end if
    end if
end while
return
```

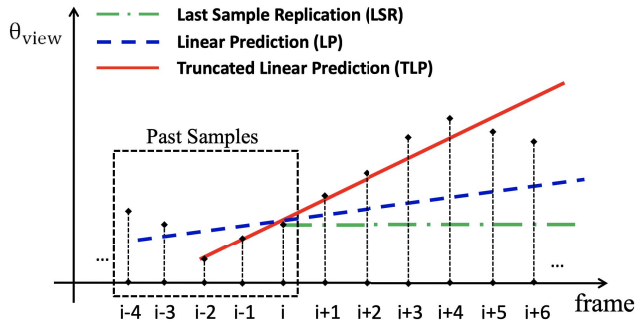


Fig. 2. View prediction methods: LSR, LP, TLP.

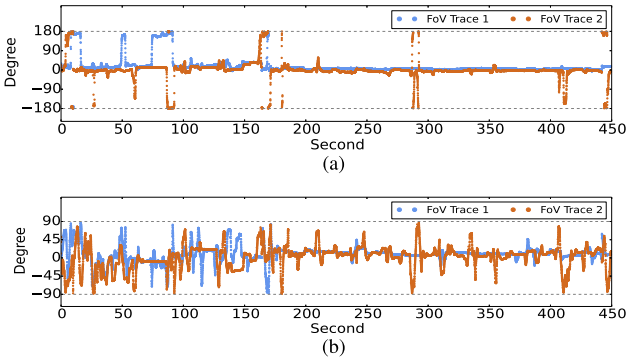


Fig. 3. Two sample FoV traces. (a) The yaw (horizontal) angle of the FoV. (b) The pitch (vertical) angle of the FoV.

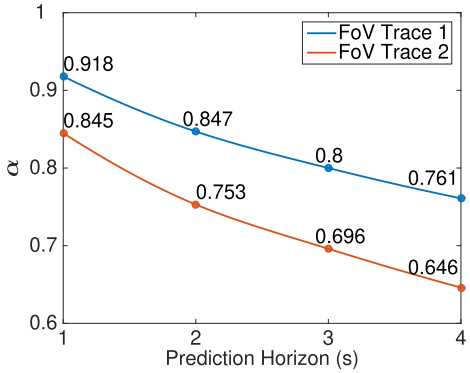


Fig. 4. FoV Hit Rate vs. prediction horizon using the truncated linear prediction method. During actual streaming, the prediction horizon at any time equals to the prefetching buffer length plus half of a chunk length.

D. Trajectory-Based FoV Prediction

A critical component of the proposed system is FoV prediction. However, chunk scheduling and system parameter optimization will work with any FoV prediction algorithm. As illustrated in Fig. 2, a simple approach is to use the view direction for the most recent frames to derive the prediction for a few frames ahead, i.e., the Last Sample Replication (LSR). To better exploit the continuity in head movement, we can use regression approach to extrapolate. One approach is linear regression based prediction (LP) illustrated by blue imaginary line in Fig. 2. In LP, all the samples covered by the imaginary box (from $i-4$ to i) are utilized for the regression. To accommodate the occasional sudden head turning, we have developed a *truncated linear prediction* (TLP) method in which we only take into account the past samples that are monotonically increasing or decreasing for extrapolation. For instance, as the red line in Fig. 2 shows, the points in the dashed window are past ground-truth samples collected for view prediction. In this case, only the last three samples inside the window (i.e., from i-2 to i) are monotonically increasing and used to make view prediction for future chunk. The three prediction schemes are illustrated in Fig. 2. The TLP algorithm is used in the simulation results presented in Sec. V. Development of more advanced FoV prediction algorithms is beyond the scope of this paper. Readers are referred to [30]–[33] for recent papers on FoV prediction.

During a streaming session, the FoV center for each future segment to be downloaded is predicted based on the FoV center of the last and several previously displayed frames. The predicted FoV center angle is then quantized (with 30° interval for each direction), and the ET viewport with the quantized FoV center is requested. In our experimental studies, each viewport covers 135° × 135° angle span for all the frames in one second. In general, due to FoV prediction error, only a portion of each decoded ET viewport will overlap with the user's FoV for all the frames in a video segment. We use the FoV hit rate, denoted by α, to evaluate the accuracy of FoV prediction, which is the overlapping ratio between the view coverage for a video segment and user's true FoVs over all frames in this segment. Figure 3 shows two sample FoV traces obtained from [34], which are the recorded traces of two users for the same 360° video. Figure 4 shows the average hit rates vs. prediction horizons for these two traces using the TLP method.

IV. Joint Optimization of Rate Allocation and ET Buffer Length

One critical design problem for the proposed two-tier system is how to allocate the rate between the two tiers and how to set up the target ET buffer length such that the rendered video quality is maximized. The optimal decision has to consider the target bit rate R_t (determined based on the available network bandwidth $\overline{B}W$), the ET chunk delivery ratio γ, and the FoV hit rate α, with the last two terms dependent on the target ET buffer length. In this section, we study the joint optimization of rate allocation and ET buffer length through a sequence of sub-problems with increasing complexity.

A. Optimal Rate Allocation for Given R_t, α and γ

We start with the basic problem of determining the optimal rate allocation with known statistics of the available network bandwidth, ET chunk delivery ratio and FoV hit rate. Adaptation of rate allocations under dynamic network environment will be considered in the following subsections.

Given the prioritized chunk scheduling algorithm, we assume that the BT chunks are always delivered before their display deadlines. Consequently, for each video segment, we either receive only the BT chunk or both the BT and ET chunks. The BT chunks are coded to cover the entire area of 360° video with the total rate of R_b (in bits/second) and the normalized video rate is therefore $\tilde{R}_b = R_b/A_b$ (bits/second/degree), where A_b is the coverage area of the 360° video, with $A_b = 360° \times 180°$. Let R_e and A_e denote the average ET rate (bits/second) and the coverage area of each ET chunk, respectively. In the following derivation, we assume the ET video and BT video are independently coded. Therefore, the normalized rate of

the ET chunk is $\tilde{R}_e = R_e/A_e$.[1] In this paper, we assume $A_e = 135° \times 135°$.

As described in Sec. III-D, we use the FoV hit rate, denoted by α, to describe FoV prediction accuracy, which is the overlapping ratio between the angle coverage for a video segment and user's true FoV over all frames in this segment, averaged over many video segments and users. It can be considered the probability that a pixel to be rendered is decodable from a received ET chunk. We further introduce γ to denote the average ET chunk delivery rate, namely the likelihood that a requested ET chunk can be successfully delivered before its display deadline. This can be determined by the ratio of the number of ET chunks that are delivered before the display time vs. the total number of chunks in a video. For a pixel in the user's FoV to be decodable from a ET chunk, it has to be within the viewport of the delivered viewport, with probability α, and the chunk has to be delivered before the display deadline, with probability γ. Therefore, the probability that a rendered pixel is covered by an ET chunk is $\alpha\gamma$. Assuming the BT and the ET coders can be characterized by their respective quality-rate (Q-R) functions (averaged over a variety of video contents and various viewports) $Q_b(\tilde{R})$ and $Q_e(\tilde{R})$, where $\tilde{R}$ is normalized bits per second per degree, the expected rendered video quality under the constraint $R_b + R_e = R_t$ can be formulated as:

$$\begin{aligned} Q(R_b; \alpha, \gamma, R_t) &= \alpha\gamma\, Q_e(\tilde{R}_e) + (1-\alpha\gamma) Q_b(\tilde{R}_b) \\ &= \alpha\gamma\, Q_e\left(\frac{R_t - R_b}{A_e}\right) + (1-\alpha\gamma) Q_b\left(\frac{R_b}{A_b}\right). \end{aligned} \tag{3}$$

With given γ and α values, the optimal R_b can be solved by setting $\frac{\partial Q}{\partial \tilde{R}_b} = 0$, which yields

$$\left.\frac{\partial Q_e}{\partial \tilde{R}}\right|_{\tilde{R}_e^*} = \beta \left.\frac{\partial Q_b}{\partial \tilde{R}}\right|_{\tilde{R}_b^*}, \quad \text{with } \beta = \left(\frac{1-\alpha\gamma}{\alpha\gamma}\right)\frac{A_e}{A_b}. \tag{4}$$

Eq. (4) implies that R_b should be chosen such that the Q-R slope at $\tilde{R}_e$ should be β times the slope at $\tilde{R}_b$. Fig. 5 demonstrates the optimal $\tilde{R}_b^*$ and $\tilde{R}_e^*$ relations for two different β values for a hypothetical but typical Q-R curve:[2] $\beta_1 = 0.03$ resulting from assuming $\alpha\gamma = 0.9$ and $A_b/A_e = 0.34$, and $\beta_2 = 0.15$ from assuming $\alpha\gamma = 0.7$. We see that if α and γ are both large, then β is very small, and the optimal allocation is to let $\tilde{R}_b$ be very low. This corresponds to the case that view and bandwidth prediction are both very accurate, so that a rendered pixel can almost always be covered by a delivered ET chunk. Under such circumstance, it is desirable not to waste bits to send entire 360° scope in the base tier. When view and/or bandwidth prediction is less accurate ($\alpha\gamma$ is lower), it is better to spend more bits on the base tier, to ensure that pixels that are rendered from BT chunks have sufficient quality. In practice, we should bound the BT rate from below as $\tilde{R}_b = \max(\tilde{R}_{b,min}, \tilde{R}_b^*)$, to ensure that any FoV region that are not covered by ET chunks due to either view prediction or delivery errors can be rendered with a basic quality of $\tilde{R}_{b,min}$.

The above analysis demonstrates that the optimal bit allocation between the two tiers depend on $\alpha\gamma$, with higher $\alpha\gamma$ leading to higher ET quality and lower BT quality. Besides the $\alpha\gamma$ factor, the optimal operation point also depends on the Q-R models for the BT and ET chunks.

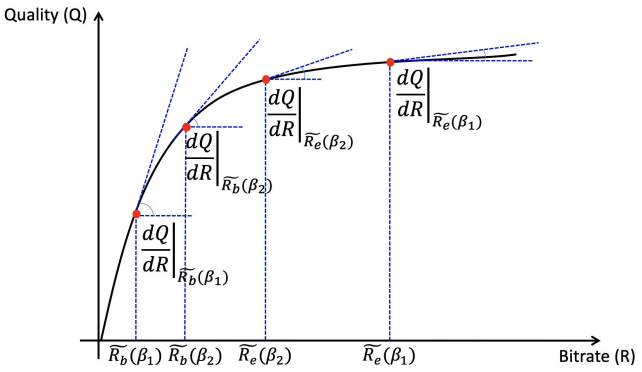


Fig. 5. Illustration of optimal rate allocation based on the Q-R slope. Two example allocations are shown, for two different β values, with $\beta_1 \le \beta_2$.

As shown in Appendix, the Q-R relation for both BT and ET chunks can be well approximated by a logarithmic function described in (11), in general with different parameters. Let a_b and b_b denote the parameters for the BT Q-R function $Q_b(\tilde{R})$, and a_e, b_e the parameters for $Q_e(\tilde{R})$. With this model, the Q-R slopes for the BT and ET are simply $b_b/\tilde{R}_b$ and $b_e/\tilde{R}_e$, respectively. Furthermore, let $\overline{BW}$ denote the average network bandwidth and η the target network utilization ratio. The target bitrate is then $R_t = \eta\overline{BW}$. The optimal rate allocation between the two tiers, represented by R_b^* and R_e^* (in bits/second) must satisfy:

$$R_t = \eta\overline{BW}, \tag{5}$$

$$R_b^* + R_e^* = R_t, \tag{6}$$

$$R_b^* = \beta' R_e^*, \quad \text{with } \beta' = \beta\frac{A_b b_b}{A_e b_e} = \frac{1-\alpha\gamma}{\alpha\gamma}\frac{b_b}{b_e} \tag{7}$$

Solving the above equations yields the optimal rate allocation solution:

$$R_b^* = \frac{\beta'}{1+\beta'} R_t, \tag{8}$$

$$R_e^* = \frac{1}{1+\beta'} R_t. \tag{9}$$

Note that with the logarithmic Q-R model in Eq. (11), the optimal rate allocation only depends on the ratio of model parameters b_b and b_e, and is independent of the parameters a_b and a_e. As shown in Appendix, Fig. 16 and Fig. 17, the ratio b_b/b_e is quite similar for two very different 360° videos, one with large motion and another one fairly stationary. Therefore, for practical implementations, one may derive the average value for this ratio from a large variety of video contents, and the optimal rate allocation does not need to be adapted based on actual video contents. This is a pleasant surprise for practical implementation of our proposed two-tier streaming system!

B. Iterative Offline Optimization of Rate Allocation and ET Buffer Length

The optimal rate allocation solution in Eq. (8,9) depends on the chunk delivery rate γ and FoV hit rate α, both dependent on the target ET prefetching buffer length B_e^T, for given chunk scheduling and FoV prediction algorithms. To prefetch an ET

[1] If layered coding is used to generate the ET chunks, relative to the coded BT chunks, the effective bitrate for ET pixels can be expressed as $\tilde{R}_e = R_b/A_b + R_e/A_e$, and the remaining derivation can be revised accordingly, as described in [4].

[2] Here we assume the $Q_b(\tilde{R})$ and $Q_e(\tilde{R})$ curves follow the same model.

chunk at a future time, one has to predict the user's FoV center at that time. In general, the longer the ET prefetching buffer (measured in video time), the lower the FoV hit rate α (see Fig. 4 for sample α curves). On the other hand, a longer prefetching buffer can better absorb network bandwidth variations, leading to higher chunk delivery rate γ, as shown in Fig. 7. From Eq. (3), it is apparent that the expected quality Q is maximal when the product $\alpha\gamma$ is maximized, for any given rate allocation. Therefore, we should choose an optimal B_e^T to maximize the product $\alpha\gamma$. However, the γ curve also depends on the actual rate allocation. Therefore the optimization of B_e^T, R_b and R_e are intertwined. We can numerically solve this complex optimization problem offline, for given bandwidth traces and FoV traces. Based on the given FoV traces and a chosen FoV prediction algorithm, we first determine the α curve by determining the FoV hit rate for each candidate prefetching time. Then we iterate between optimizing rate allocation and deriving the γ curve. Given an initial rate allocation (we used $R_b = 0.2\ R_t$ and $R_e = 0.8R_t$, and we generate three possible ET rates with $R_{e1} = 0.8R_e$, $R_{e2} = R_e$ and $R_{e3} = 1.2R_e$), streaming simulations using given bandwidth traces are repeated with different target buffer lengths B_e^T (ranging from 1 to 4 seconds). For each B_e^T, the average γ is calculated. The γ values for all the possible B_e^T make up the γ curve for the current rate allocation. The buffer length that maximizes the product $\alpha\gamma$ is chosen as the target buffer length for the next iteration. Optimal rate allocation is determined based on the maximum value of $\alpha\gamma$ by solving Eq. (6) and (7). We then start the next iteration using the updated B_e^T, R_b and R_e rates. This process continues until the solution converges or the maximum number of iteration is reached. In our simulations, convergence is reached usually within four iterations.

The optimization of the rate allocation and ET buffer length described above requires the knowledge of the dynamics of network bandwidth and user's viewing behavior. The former is characterized by the average bandwidth $\overline{BW}$ and the chunk delivery rate γ as a function of target ET buffer length B_e^T for a given rate allocation. The latter is described by the FoV prediction hit rate α as a function B_e^T. To operate the iterative optimization offline, we assume typical bandwidth traces and FoV traces can be collected, which reflect the expected network and FoV dynamics, from which $\overline{BW}$ and the α and γ curves can be calculated.

However the actual bandwidth and FoV dynamics in a particular streaming session may be quite different from what are assumed for the static optimization, and in fact these dynamics may change substantially within the same streaming session. In practice, one should dynamically estimate $\overline{BW}$ and γ and α functions based on the observed network and user FoV dynamics in the recent past, and use them to adapt the ET buffer length and rate allocation. We assume that the average bandwidth $\overline{BW}$ can be estimated fairly accurately by averaging the throughputs for downloading previous chunks.

We will first study how to dynamically update the γ and α functions in Sec. IV-C and Sec. IV-D. We then present two online optimization methods: *periodic optimization* and *adaptive optimization* in Sec. IV-E and IV-F, respectively.

C. Online Estimation of γ Function

In operational networks, the bandwidth available to a video streaming session is subject to various interferences, such as radio interference on a wireless link or time-varying cross traffic injected by other competing flows sharing the same wired bottleneck link, etc. All these factors not only change the average bandwidth for the session, but also introduce different levels of bandwidth variations to it. For example, Fig. 6 shows the bandwidth variations of five sessions over a WiGig link subject to different levels of blockages. The blockage level gradually increases from session 1 to 5. The detailed statistics about the bandwidth traces can be found in Table I.

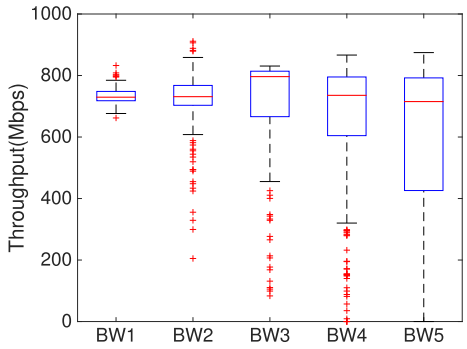


Fig. 6. Bandwidth variations of five sessions over a WiGig link under different levels of blockages.

TABLE I
5G WIRELESS TRACES INFORMATION

Traces	Throughput (Mbps)				
	Mean	SD	Max	Min	Med
BW 1	734.34	22.51	832.50	662.0	729.5
BW 2	722.80	85.92	911.40	205.05	731.20
BW 3	719.03	154.81	830.80	83.37	796.40
BW 4	659.67	206.73	866.50	0	735.50
BW 5	585.31	277.10	874.5	0	715.25

Dynamically estimating the γ function is hard because video streaming session operates with a specific target ET buffer length at any given time, so we can only get the γ value for the current buffer length. Therefore, we need a way to estimate the γ function for the entire range of B_e^T on the fly.

To understand how the average bandwidth $\overline{BW}$ and bandwidth variation pattern affect the γ functions, we have evaluated these functions for various bandwidth traces as well as their scaled and shifted versions. In these simulations, we choose the BT and ET rate R_b and R_e so that $R_b + R_e = \eta\overline{BW}$ and we fix η to a conservative value, i.e., $\eta = 0.85$.

Through these evaluations, we have identified some interesting patterns of the γ function:

- If the average bandwidth becomes larger or smaller but the relative bandwidth variations remain at the same level, the resulting γ function typically remains the same as long as we use the same utilization η for the rate allocation. This means that, with a conservative network utilization ratio, the γ curve will not be significantly affected by the individual R_b and R_e values, nor the average bandwidth $\overline{BW}$.
- When the relative bandwidth variation changes dramatically, the γ function could be much different. For example, when the average bandwidth is the same, but bandwidth fluctuates with a larger standard deviation, the γ value increases much slower with the ET buffer length.

From the above observations, we have decided to use the relative standard deviation of bandwidth, c_v, to characterize the bandwidth dynamics, with $c_v \triangleq \sigma/\mu$, where σ is the

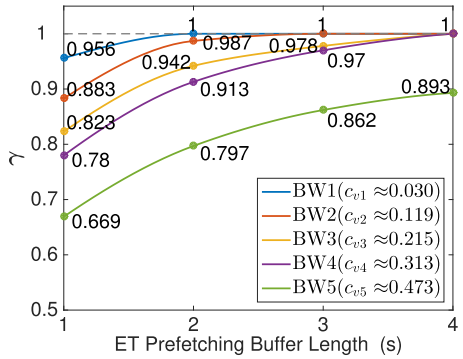


Fig. 7. ET chunk delivery rate for networks with different dynamics.

standard deviation and μ is the mean value of the bandwidth over the observed period. The γ curves for the five network traces summarized in Table I are illustrated in Fig. 7 with their corresponding c_v values. Even though we conclude that γ curve is not significantly affected by the rate allocation, to ensure the γ curves can represent network traces perfectly, for each individual network trace, we generate multiple γ curves under different rate allocations and use the average among these curves. We precompute and store a set of γ functions for a discrete set of c_v values through offline simulations. When a streaming session starts, we measure the instantaneous bandwidth when downloading each BT/ET chunk. The average bandwidth and the standard deviation are dynamically updated using the instantaneous bandwidth measurement over the past T chunks. Then at the beginning of each adaption interval, we calculate the relative standard deviation for the last time period and choose the stored γ function with the closest relative standard deviation as the current estimate of γ function.

D. Online Estimation of α Function

Similarly, the α function depends on the FoV dynamics of users and the video content. We need to dynamically estimate the current α function as the streaming session progresses. In order to update the α function, FoV prediction for the following k seconds is conducted after the display of each second of video. For example, assuming the current time is t. New FoV prediction P_t for ($t+1$ to $t+k$) is generated. We name each value in prediction P_t as $P_{t,i}$ and i ranges from 1 to k which is represented as the bottom part in Fig. 8 ($k = 4$). After displaying video of each second (with the actual user FoV), the FoV hit rate for previous FoV predictions can be computed. As illustrated in Fig. 8, at time t, FoV hit rate of all the previous FoV predictions from $t-1$ to $t-k$, including $P_{(t-1),1}$, $P_{(t-2),2}$, $P_{(t-3),3}$, $\cdots$, $P_{(t-k),k}$, are calculated. All the gray shadow areas represent FoV predictions that have been already validated and predictions without validation are represented as boxes of blue slash. So at time t, we can update the $\alpha(T_p)$ by averaging the hit rates for P_{t-i,T_p}, for those predictions that have been validated over past T seconds.

E. Periodic Online Optimization of Rate Allocation and ET Buffer Length

In this approach, we periodically update the average bandwidth $\overline{BW}$, the γ and α functions and determine the optimal solution for the rate allocation and ET buffer length based on the updated information following Eq. (6) and (7).

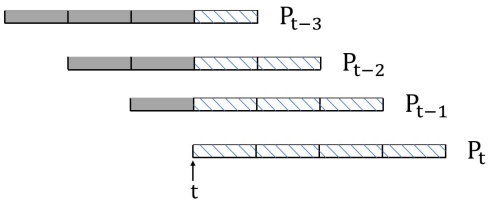


Fig. 8. FoV accuracy function estimation through progressive FoV prediction and validation.

We set the initial rate allocation and target ET buffer length based on some expected average bandwidth, γ function and α function ($R_b = 80$Mbps, $R_{e1} = 450$Mbps, $R_{e2} = 550$Mbps, $R_{e3} = 650$Mbps and B_e^T is set to 2s). After working with this initial setting for a period of T_1, a new optimization is triggered. The α and γ functions are updated using measurements in the previous T_1 seconds, following the procedures outlined in Secs. IV-C and IV-D.

In practice, we only code the BT and ET videos at a finite set of discrete rates, and the optimal rates will be quantized to their nearest rate neighbors in this finite set. Furthermore, the optimal ET rate represents the average ET rate, and several adjacent ET rates in the finite set will be chosen so that their average is close to the optimal ET rate. As described earlier, the ET video can be coded independently, or relative to the BT video using layered coding, to improve the system efficiency. However, when using dynamic rate adaptation, layered coding causes additional complexity. First of all, if layered coding is used, the server has to generate and store a different set of ET chunks for each possible BT rate. Furthermore, as we have two independent buffers for BT and ET, respectively, at the beginning of each adaption, there might be prefetched BT and ET video chunks in the buffers. If the newly selected rate set for BT and ET are different from the previous one, the newly download ET chunks cannot be decoded with the previously downloaded BT chunks for the same video time. Using non-layered coding will completely avoid such problems, making it more attractive for practical systems with dynamic adaptation. For this reason as well as other reasons explained in Sec. III-B(b), simulation results presented in Sec. V-E are all based on non-layered coding.

F. Adaptive Online Optimization of Rate Allocation and ET Buffer Length

In 360° video streaming, the network status and the user's viewing behavior could change suddenly at random time. Periodic adaption may not be able to adapt fast enough to the sudden changes on the one hand, and may waste a lot of computation resources when the environment is relatively stable on the other hand.

To cope with this, we also propose an *adaptive optimization* approach. In this case, adaption is triggered by significant changes in the network status or FoV direction. In particular, $\overline{BW}$, c_v and α curve are updated after downloading each chunk based on the measured bandwidths and FoV prediction accuracy for the past T_2 chunks. A threshold on the relative change is preset to determine whether the change in any one of $\overline{BW}$, c_v or α is significant enough to trigger a new optimization. The threshold value is an important system parameter to be tuned, as improper setting of the threshold might lead to an either oscillating or unresponsive system. Based on our experience, the threshold is set to 10% for

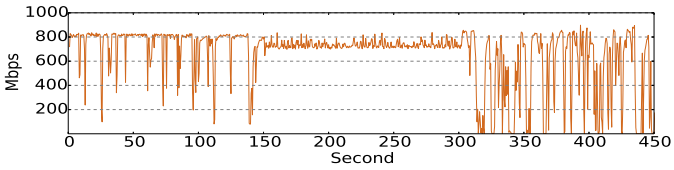


Fig. 9. Concatenated bandwidth trace with different levels of dynamics.

$\overline{BW}$ and 0.1 for c_v. As α values for four different prediction intervals are evaluated in realtime, we calculate the Euclidean distance d_α between current α curve and the one for last period. Threshold for d_α is set to 0.2.

Note that we choose not to trigger the optimization based on the change in short-term FoV prediction accuracy, because the FoV prediction accuracy is not very stable. However, we still continuously update the α function as described in Sec. IV-D, and uses the latest α function in the newly triggered optimization. The same initial setting of rate allocation and target ET buffer length with Sec. IV-E is used for adaptive optimization.

V. Experimental Evaluation

We conduct trace-driven simulations of the proposed two-tier streaming system with dynamic adaptation of system parameters, and compare it with several benchmark systems. Our experiments mainly consider the very high-bandwidth 5G wireless networks, which has the potential to deliver high resolution 360° video (e.g. 24K×12K and 120 Hz). We also show the results for low-bandwidth LTE networks, to demonstrate that the proposed framework is efficient for both scenarios.

A. Bandwidth and FoV Traces

1) WiGig Bandwidth Traces: We perform simulations using five WiGig bandwidth traces collected by transmitting data via TCP in *iperf*. We choose to examine the performances with 5G networks because they afford high bandwidths necessary for high quality 360° video streaming. Blockages by different materials, e.g., metal, human body, or books, are introduced to interrupt the transmission with different durations to emulate networks with different levels of dynamics. The bandwidth evolution of the five traces are summarized in Fig. 6 and Table I. To simulate a streaming session with time-varying dynamics, we concatenate three network traces with very different dynamics to create a synthesized trace shown in Fig. 9. The 450s bandwidth trace is composed of parts of the three original bandwidth traces. The first 150 seconds is from Trace 3 with medium bandwidth variations, the middle 150 seconds is from the stable Trace 1, and the last 150s is from the most fluctuating Trace 5.

2) FoV Traces: We use two FoV traces for two different videos in the public dataset [34]. Fig. 3 shows the FoV records for the horizontal and vertical directions.

B. System Parameter Setting

For the two-tier system, the BT chunk view coverage is 360° × 180°, ET view coverage is 135° × 135°. We assume the user FoV for any frame is 105° × 105° and we set the ET span to be larger than the FoV to accommodate FoV changes within the duration of a chunk and the likely FoV prediction errors. Video chunk duration is set to 1 second and the total video length is 450 seconds, the same length as the synthesized network trace. The value of k_p and k_i in *P-I* controller are 0.6 and 0.01, respectively, and buffer length record for the past 10 seconds is utilized by *P-I* controller. For rate allocation optimization, the candidate target ET buffer lengths range from 1 to 4 seconds, ET buffer length upper bound is 2 seconds larger than the chosen target buffer length.

TABLE II
The Set of Discrete Rates (Mbps) Used by the Two-Tier System

Base Tier	10, 30, 50, 80, 120, 160, 200, 250
Enhancement Tier	300, 350, 400, 450, 500, 550, 600, 650, 700

Given the high throughput of 5G networks (e.g., up to 800 Mbps), our experiments target at very high resolution 360° videos (e.g., 24K in resolution, 90-120 Hz, and 12-bit video source). Instead of actually generating the video bitstreams, we assume the videos can be coded into a set of BT rates and ET rates, as summarized in Tab. II. These rates are chosen to cover the bandwidth range of the synthesized bandwidth trace. We also assume accurate encoder rate control, such that each BT or ET chunk can be coded exactly at one of the rates defined in Tab. II.

For periodic optimization, the update interval is $T_1 = 30$ seconds. The average bandwidth $\overline{BW}$, relative standard deviation c_v and FoV hit rate curve $\alpha(B_e^T)$ are calculated based on the records for the past 30 seconds. For the adaptive optimization, the observed bandwidth in the past $T_2 = 10$ seconds are used to compute $\overline{BW}$ and c_v, and optimization will be triggered if $\overline{BW}$ increases or decreases by 10% or the absolute change in c_v is more than 0.1 or if the change in α curve d_α exceeds 0.2.

C. Benchmark Systems

We compare the proposed two-tier system with dynamic optimization with the following benchmark systems:

- **Naive 360° DASH Streaming**: This system covers the entire 360° video content in each video chunk. Each 360° video segment is precoded into several available rates ranging from 100Mbps to 850Mbps with gap of 50Mbps. The chunk scheduling is accomplished by a *P-I* controller based algorithm for 2D planar video streaming [29] with a target buffer length of 10 second and buffer upper-bound of 20 seconds.
- **Predictive Single-Tier Streaming**: In this system, the client predicts the user FoV for the video segments to be requested and requests the chunks covering the predicted FoV. This system is a special case of the two tier system but without the base tier. The chunks cover 135° × 135° view span and are coded into multiple rates, ranging from 100Mbps to 850Mbps with gap of 50Mbps. Obviously, due to FoV prediction error, "black" regions may appear when users suddenly change their view directions. We also use the *P-I* controller approach [29] to schedule the chunk request. We tried different target buffer lengths, and found that using 3 seconds leads to the highest QoE for both FoV traces.
- **Static Two-Tier Streaming**: In this case, target ET buffer length and rate allocation are fixed throughout the streaming session (450s). We present results from two different settings. The first setting uses the network trace and the FoV trace in the first 150 seconds to determine the optimal operating parameters for the entire 450 seconds streaming session (Static 1). The second setting uses the entire bandwidth and FoV traces to determine the optimal parameters (Static 2). Note that either approach is not

practical, but we include results from these two cases to examine the potential gain from dynamic adaptation.

D. Quality of Experience Metric

To evaluate and compare the performances of various systems, we consider three primary factors affecting the users' quality of experience. i) Average rendered video quality for pixels that arrive before display time and are within user's FoV, to be denoted by Q_r. ii) The influence of video freezing due to late arrival of some chunks. We use Q_f to denote the perceived quality due to freezing, which we assign a negative number to emphasize its negative impact. We use p_f to denote the percentage of times during an entire streaming session that freezing occurs. iii) In general, the decodable area from the received chunks (before display deadline) for a video segment may not completely cover user's FoV over all frames in this segment. The missing region may be rendered "black". We use Q_b to denote the perceived quality resulting from these "black" pixels, which we also assign a negative number. We use p_b to denote the percentage of pixels during an entire streaming session that are black. Note that with the two-tier system, because the base-tier covers the entire 360° span, "black" regions will not occur, as long as the BT chunks are received in time. When the BT chunks are late, they will lead to "freezing". However, with the benchmark single-tier system, because delivered chunks only cover the predicted FoV, "black" region will appear if the prediction is inaccurate, which could happen when users change FoV direction suddenly. Overall, we define the QoE over a streaming session as

$$QoE = (1-p_f)(1-p_b)Q_r + p_f Q_f + (1-p_f)p_b Q_b \quad (10)$$

To determine the rendered video quality Q_r, we average the quality over all the displayed segments in non-black regions. With our two-tier system, a rendered pixel may be decoded from either an ET or BT chunk. The quality for video segment n when both the ET and BT chunks are available can be expressed as

$$Q_{r,n} = \alpha_n Q_e\left(R_{e,n}/A_e\right) + (1-\alpha_n) Q_b\left(R_{b,n}/A_b\right),$$

where α_n is the FoV hit rate of the ET chunk for this segment (i.e., the percentage of pixels covered by the ET chunk) and $R_{e,n}$ and $R_{b,n}$ are the rates (in bits/second) of the ET and BT chunks, respectively. The quality when only the BT chunk is available is

$$Q_{r,n} = Q_b\left(R_{b,n}/A_b\right).$$

Note that by averaging over all chunks, Q_r is equivalent to the expected quality described in Eq. (3).

The naive 360° DASH system and the single-tier system can be considered special cases of the two-tier system without the ET or BT chunks, respectively. For the naive 360° system, for the chunks arrived before the display deadline with rate R_n, $Q_{r,n} = Q_b\left(R_n/A_b\right)$. For the single-tier system, $Q_{r,n} = Q_e\left(R_n/A_e\right)$. We assume the perceived quality for rendered pixels in a FoV is logarithmically related to the normalized bitrate of the rendered pixels, as in Eq. (11). Instead of performing actual coding to determine parameters a and b for the Q-R function, which is not feasible as we do not have access to 24K 360° video, we determine these parameters by letting the quality at a large rendering bitrate $\tilde{R}_{\max}$ to be 10, and the quality at a small rendering rate $\tilde{R}_{\min}$ to be 0. Specifically, we set $\tilde{R}_{\max} = 700\times10^3 \text{Kbps}/A_e$, $\tilde{R}_{\min} = 10\times10^3 \text{Kbps}/A_b$.

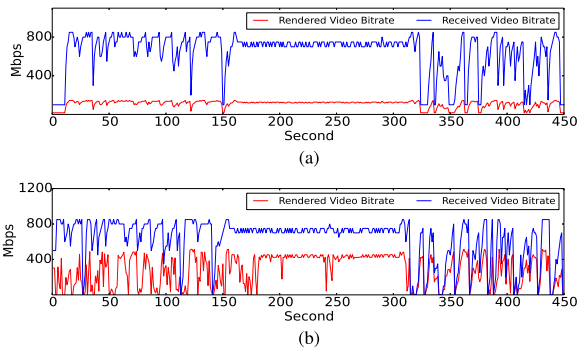


Fig. 10. Received and rendered video rate of benchmark systems. (a) Naive 360° DASH Streaming under WiGig. (b) Predictive Single-Tier Streaming under WiGig.

This yields $a = 3.72$ and $b = 1.72$. We set the "perceived" quality during freezing and for "black" region to the same negative value, i.e., $Q_f = Q_b = -1$. The same Q-R function is used for all the comparison systems. Note that because our coding experiments in Appendix demonstrate that the parameters of the Q-R functions for the BT and ET chunks are quite similar, for simplicity, we apply the same Q-R model parameters for the BT and ET chunks in the experimental study.

E. Performance Comparison in WiGig Network

We compare the performances of the proposed two-tier streaming system with two different dynamic optimization strategies and two benchmark systems. For naive 360° and predictive single-tier streaming, the received and rendered video bitrates are illustrated in Fig. 10. Received bitrate is defined as the total bitrate of the received chunks for that particular time and rendered bitrate is the rate rendered within user FoV. In naive 360° system, even though the received video bitrate is high, the effective rendered video rate is only about 17.01% of it; on the contrary, about 60.5% of the received video rate is effective in single tier system as long as the FoV prediction is accurate. However, single-tier system is not robust and stable enough due to the fluctuating bandwidth and sudden user FoV change. The performance of the static two-tier is similar as [4], so no detailed figures are shown.

For each of the dynamic optimization strategies, three metrics are illustrated: buffer length, video bitrate and rate allocation. Fig. 11 illustrates the performance with periodic optimization. As shown in Fig. 11(a), with periodic optimization, BT buffer length mostly stays at a safe level except for at time of 300s when bandwidth migrates from stable to fluctuating. As expected, the ET buffer length varies with the network environment and FoV dynamic, and follows the target ET buffer length determined by dynamic optimization. In the first two phases, the ET buffer is around 1-2 seconds due to the dynamic FoV traces. The purpose is to improve α as value of γ can be guaranteed even with a short buffer length. During the third phase, the target buffer length increases to 3 or 4 seconds to adapt to the fluctuating bandwidth. However, the ET buffer still runs out for some periods of time. Fig. 11(b) illustrates how the delivered and rendered video rate changes in time. Blue curve represents the bitrate of the received video chunks; Red curve is the bitrate of video that can be potentially rendered to the user without considering missing deadline ($\gamma = 1$) and FoV prediction error ($\alpha = 1$); Green curve is the eventual rendered bitrates for the user. The received bitrate is

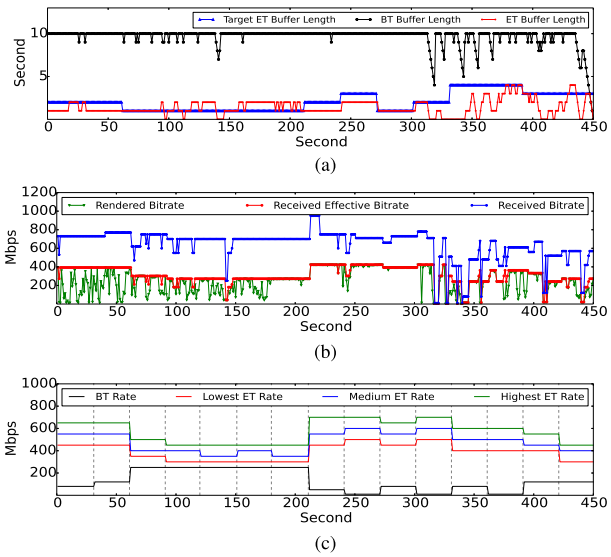


Fig. 11. Performance of periodic optimization under WiGig network trace and FoV trace 1. (a) Buffer length. (b) Video rate. (c) Rate allocation.

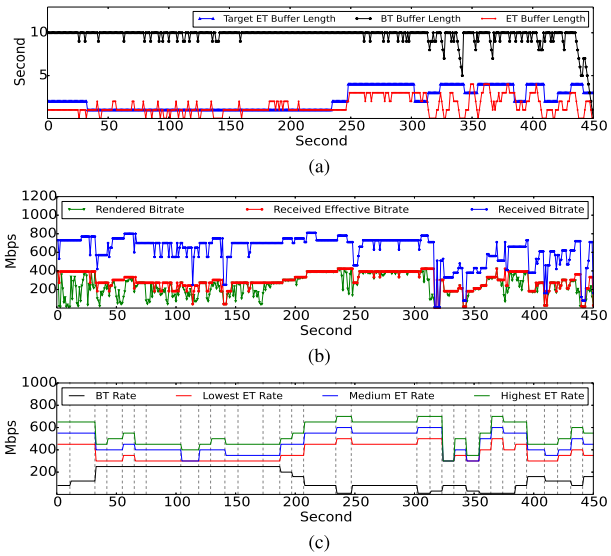


Fig. 12. Performance of adaptive optimization under WiGig network trace and FoV trace 1. (a) Buffer length. (b) Video rate. (c) Rate allocation.

mostly stable, the reason why the rendered bitrate is lower is either FoV prediction is not accurate (α is low) or ET chunk is not delivered in time (γ is low), or both. Different from the first two phases, even the received bitrate becomes fluctuating in the third phase, due to the oscillating network environment. To show how dynamic optimization is operated, the quantized optimal rates for BT and ET chunks after each optimization are shown in Fig. 11(c). After each optimization (indicated by the gray lines), BT and ET rates will be adjusted based on historical information. For instance, during time of 0 to 170 seconds, FoV hit rate α is low, a significant portion of the available rate is assigned to the BT; however, when the FoV prediction is accurate, e.g., around 400 seconds, the system allocates most rate to ET.

For adaptive optimization, both buffer lengths and video rates perform similarly to periodic optimization. However, Fig. 12(c) shows that the optimization of the rate allocation and target ET buffer length is remarkably different from the periodic optimization. Adaptive optimization is only triggered

TABLE III

PERFORMANCE UNDER WIGIG NETWORK TRACE AND FOV TRACE 1

Metrics	Strategies					
	Naive 360	Single Tier	Two Tier System			
			Static 1	Static 2	Periodic	Adaptive
Ave QoE	7.03	7.33	7.55	7.50	7.57	7.84
Rendered Quality Q_r	7.03	7.71	7.55	7.50	7.57	7.84
Ave Display Rate (Mbps)	110.27	281.94	236.73	222.03	242.83	244.86
Ave Received Rate (Mbps)	648.11	627.56	646.89	609.44	641.56	631.04
Freezing Ratio	0	5.78 %	0	0	0	0
Average Black Ratio	0	25.31 %	0	0	0	0
Number of Optimizations	0	0	1	1	15	32

TABLE IV

PERFORMANCE UNDER WIGIG NETWORK TRACE AND FOV TRACE 2

Metrics	Strategies					
	Naive 360	Single Tier	Two Tier System			
			Static 1	Static 2	Periodic	Adaptive
Ave QoE	7.03	6.67	7.23	7.13	7.36	7.42
Rendered Quality Q_r	7.03	7.15	7.23	7.13	7.36	7.42
Ave Display Rate (Mbps)	110.27	239.59	175.24	176.82	184.62	180.94
Ave Received Rate (Mbps)	648.11	627.56	635.78	602.22	632.33	620.55
Freezing Ratio	0	5.78 %	0	0	0	0
Average Black Ratio	0	36.87 %	0	0	0	0
Number of Optimizations	0	0	1	1	15	31

TABLE V

TWO TIER SYSTEMS DETAIL UNDER FOV 1

Metrics	Static 1	Static 2	Periodic	Adaptive
Ave BT Rate (Mbps)	120.0	120.0	127.77	139.37
Ave ET Rate (Mbps)	526.89	489.44	513.78	491.67
Ave Fov Hit Rate α	0.75	0.74	0.77	0.81
Ave Chunk Delivery Rate γ	0.87	0.89	0.92	0.94

TABLE VI

TWO TIER SYSTEMS DETAIL UNDER FOV 2

Metrics	Static 1	Static 2	Periodic	Adaptive
Ave BT Rate (Mbps)	200.0	160.0	207.55	218.22
Ave ET Rate (Mbps)	435.77	442.22	424.77	402.33
Ave Fov Hit Rate α	0.63	0.63	0.68	0.70
Ave Chunk Delivery Rate γ	0.87	0.89	0.88	0.87

by specific events. For example, during the second phase (200 to 300 seconds), optimization is seldom triggered because the network bandwidth and FoV direction are almost constant. Meanwhile, optimization is operated almost every 10 seconds in the third phase.

Tables III and IV compare the overall QoE and individual performance metrics of all systems. In all cases, the four two-tier systems outperform the benchmark systems, and the adaptive optimization is better than periodic optimization, which is better than static optimization (even when statistic optimization is based on the network statistics calculated from the underlying network trace). For FoV trace 1, which is less dynamic, the single tier system is better than the naive system, because FoV prediction is mostly accurate. However for FoV trace 2, the single-tier system performs the worst among all the systems. Except for the single-tier system, no video freezing and black screen occurs. It further shows that single-tier streaming is not stable. While comparing performances between the two tables, as expected the two-tier systems have better performance with the more stable FoV trace 1.

Tables V and VI show the detailed information about the four two-tier systems. The two dynamic optimization systems outperform the two static system in terms of FoV hit rate α for both FoV traces and chunk delivery rate γ for FoV trace 1, as the target buffer length is adjusted in real-time. From the comparison between the two tables, we find BT is allocated with higher rate when α value is low. For each system, γ value are roughly the same for different FoV traces, even though BT and ET are allocated with different rates.

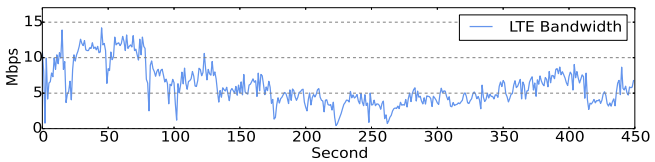


Fig. 13. Dynamic LTE bandwidth trace.

TABLE VII

THE SET OF DISCRETE RATES (MBPS) USED BY THE TWO-TIER SYSTEM

Base Tier	0.1, 0.3, 0.5, 0.7, 1.0, 1.5, 2.0, 2.5
Enhancement Tier	3.0, 4.0, 5.0, 7.5, 10.0, 12.5, 15.0

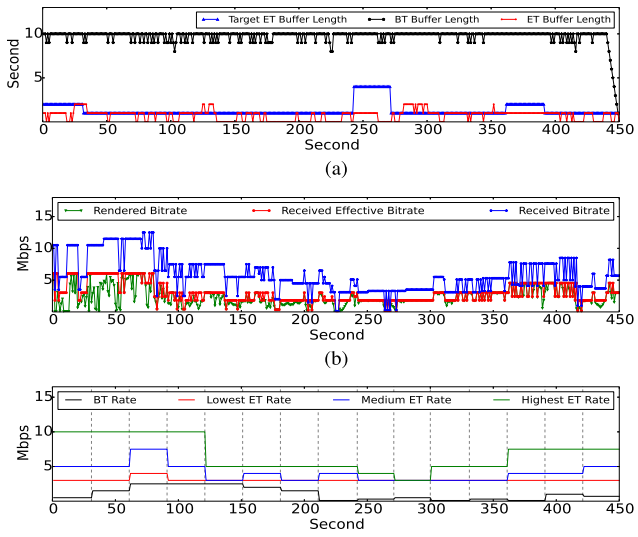


Fig. 14. Performance of periodic optimization under LTE network trace and FoV trace 1. (a) Buffer length. (b) Video rate. (c) Rate allocation.

F. Performance in LTE Network

To demonstrate that the proposed system also works in other dynamic networks, we conduct experiments similar to Sec. V-E on LTE bandwidth traces collected by a LTE mobile phone on a bus, illustrated in Fig. 13. We find that LTE bandwidth evolution is quite different from WiGig. Therefore, we manually select four typical LTE traces, and compute γ curve offline using the same method as Sec. IV-C. The corresponding BT and ET rates are pre-defined in Table VII. Because the bandwidth range for LTE is relatively large, for the static variant of the two-tier system, once the optimal rate R_e is determined, we generate three ET rates as: $R_{e1} = 0.5R_e$, $R_{e2} = R_e$ and $R_{e3} = 1.5R_e$. We use the FoV trace 1 to evaluate the two-tier system performance in dynamic LTE network using the trace shown in Fig. 13. The Q-R model parameters are chosen so the the quality ranges from 0 to 10 in the normalized rate range corresponding to the bit rate range in Table VII, which yields $a = 6.34$ and $b = 1.517$.

Both periodic and adaptive optimization strategies are tested, and the results are demonstrated in Fig. 14 and Fig. 15, respectively. As expected, BT buffer length is stable without freezing, and ET buffer length varies with the target ET buffer length generated from optimization. The received and rendered video bitrates shown in Fig. 14(b) demonstrate that during the period from 30s to 75s, with relatively high available bandwidth, the highest ET rates are delivered in most cases. However, due to the dynamic FoV change, even though high bitrate chunks are available, the eventual rendered bitrates are still low. And this further explains why target ET buffer length keeps at one second within this time period. Fig. 15(b) illustrates the adaptive strategy behaves in a similar way. In LTE network, the adaptive strategy conducts one rate optimization almost every 10 seconds to adapt to the dynamic LTE bandwidth and FoV change, and results in 41 optimizations in total during the entire 450 seconds, which is more frequent than the periodic one. As a consequence, the rate allocation with adaptive strategy changes much smoother.

TABLE VIII

PERFORMANCE UNDER LTE NETWORK TRACE AND FOV TRACE 1

Metrics	Strategies				
	Naive 360	Single Tier	Two Tier System		
			Static 2	Periodic	Adaptive
Ave QoE	6.06	6.33	6.33	6.73	6.73
Rendered Quality Q_r	6.06	6.65	6.33	6.73	6.73
Ave Display Rate (Mbps)	0.99	2.59	2.12	2.25	2.22
Ave Received Rate (Mbps)	5.80	5.50	5.49	5.84	5.75
Freezing Ratio	0	7.33 %	0	0	0
Average Black Ratio	0	17.64 %	0	0	0
Number of Optimizations	0	0	1	15	41

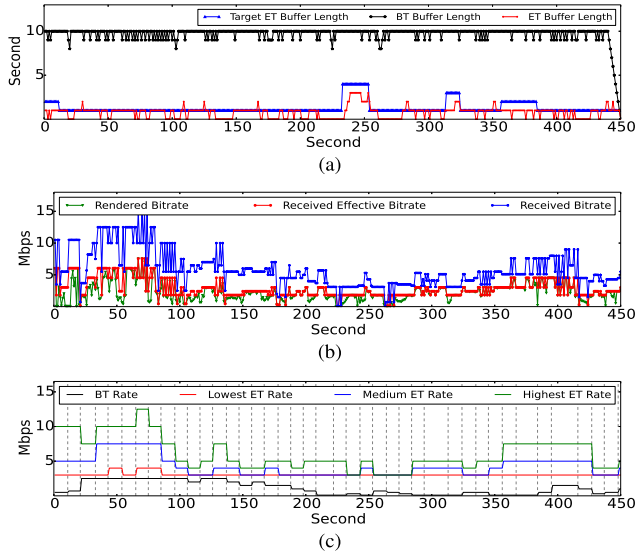


Fig. 15. Performance of adaptive optimization under LTE network trace and FoV trace 1. (a) Buffer length. (b) Video rate. (c) Rate allocation.

We also compare the performance among different streaming strategies. In Table VIII, Naive 360° and single-tier have 13 available bitrates ranging from 0.1 to 15.0Mbps. All the three two-tier systems share the same initial and streaming setting. The results demonstrate that the adaptive approach that has frequent rate optimization generates the highest QoE, and followed by the periodic optimization.

VI. CONCLUSIONS

In this paper, we developed a novel two-tier 360° video streaming framework to maximize the rendered video quality, while maintaining the streaming continuity and robustness against the inherent dynamics in both user FoV and network bandwidth. We analytically studied the optimization of the target ET buffer length and rate allocation between the BT and ET. We further developed algorithms that dynamically adjust the rate allocation and ET buffer length based on the real-time measurement of the network bandwidth statistics and FoV prediction accuracy. Through experiments driven by real 5G 802.11ad and LTE bandwidth traces and real user FoV traces, we demonstrated that the proposed two-tier

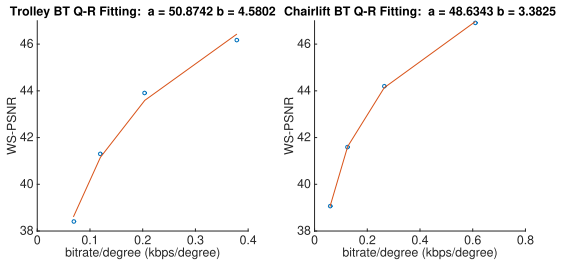


Fig. 16. Base-tier WS-PSNR vs. normalized rate curves for two test sequences.

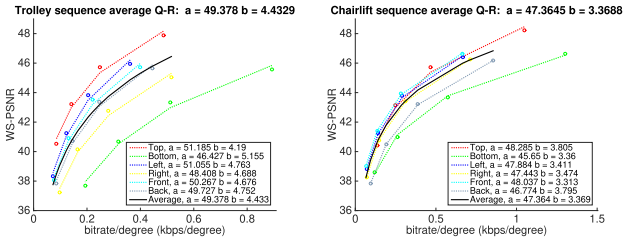


Fig. 17. Enhancement-tier WS-PSNR vs. normalized rate curves for different viewports.

systems substantially outperform the regular DASH streaming and single improvement in quality of experience over two benchmark systems.

APPENDIX
VIDEO CODING EXPERIMENTS AND QUALITY-RATE MODELING

To quantify the differences in the achievable quality-rate (Q-R) performances between the BT and ET, and the influence of the viewport directions on the ET coding efficiency, we have conducted coding experiments as follows. We choose two 8K JVET sequences: "Trolley" (8K-8bit), a static scene captured by fixed camera and "Chairlift" (8K-10bit), a moving scene captured by a mounted camera, both represented in the ERP format with 8192×4096 pixels and 30 frames/second.

For the BT, we code the first 30 frames of each ERP video using the HEVC reference software (HM) [35], following the JVET common test condition (CTC), using four quantization parameters (i.e., 22, 27, 32, 37) in random access (RA) configuration. For the lack of well accepted subjective quality metrics for 360° video, we will assume the perceptual quality is proportional to the weighted-to-spherically-uniform peak-signal-to-noise ratio (WS-PSNR), which is a 360° video objective metric recommended by JVET, in which the geometrical distortion of ERP is taken into account by assigning different weights to different pixel locations in the ERP [36]. The WS-PSNR vs. normalized rate points corresponding to different QPs are shown in Fig. 16, where the normalized rate has a unit of Kbits/second/degree, determined by dividing the total bitrate (in kbits/second) by the total degree area covered by the BT, which is $A_b = 360° \times 180°$. We have found that these Q-R curves can be represented quite well by a logarithmic model

$$Q(\tilde{R}) = a + b \cdot \log \tilde{R}, \tag{11}$$

where a and b are content-dependent. We would like to note that the logarithmic Q-R relationship has been widely observed for 2D video, when the quality is evaluated by PSNR. The fact that this is also true for 360° video when the quality is measured by WS-PSNR is thus not a coincidence.

For the ET, we assume each viewport covers a view span of $135° \times 135°$. To realize tile-based coding, we divide the entire ERP region into 16×8 tiles, each with 512×512 pixels. We code each tile using the same configuration as for the BT. For each viewport, we determine all the tiles needed to cover the viewport by projecting the FoV corresponding to each viewport center back to the ERP. We evaluate the total rate and the average WS-PSNR for each QP. The resulting WS-PSNR vs. normalized rate $\tilde{R}$ for six viewport directions (front, back, left, right, top, and bottom) are shown in Fig. 17. The normalized rate is determined by dividing the total bitrate (in Kbits/second) by the total degree area covered by an ET chunk, which is $A_e = 135° \times 135°$.

We see that the Q-R curves for the middle viewports are quite similar, but the Q-R curves for the top and bottom viewports are quite different, and furthermore the top view can be coded more efficiently than the other views, as expected. Figure 17 shows that Q-R curves for different viewports can also be fitted very well by the logarithmic model, and the parameters "a" and "b" generally are viewport dependent. However the "b" values for different viewing directions are relatively close.

Although the Q-R curves for the top and bottom viewports are quite different from those for the middle viewports, the probability that a user will look at the top and bottom directions is relatively small. From the FoV trace data [34], we have found that the total probability that the latitude direction of the FoV center fall in the range of $\pi/4$ to $\pi/2$ and $-\pi/2$ to $-\pi/4$ is less than 10%. Therefore, we can safely use the WS-PSNR vs. rate points corresponding to the middle view directions to derive the average Q-R curve, which is also shown in Fig. 17. Comparing the Q-R curves for the BT and ET viewports, we see that BT coding is more efficient. This is as expected as the entire ERP is coded together. However, the difference between the model parameters for the same video is relatively small.

REFERENCES

[1] Huawei. (2016). *Whitepaper on the VR-Oriented Bearer Network Requirement*. [Online]. Available: http://www-file.huawei.com/~/media/CORPORATE/PDF/white%20paper/whitepaper-on-the-vr-oriented-bearer-network-requirement-en.pdf

[2] F. Duanmu, E. Kurdoglu, Y. Liu, and Y. Wang, "View direction and bandwidth adaptive 360 degree video streaming using a two-tier system," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[3] F. Duanmu, E. Kurdoglu, S. A. Hosseini, Y. Liu, and Y. Wang, "Prioritized buffer control in two-tier 360 video streaming," in *Proc. Workshop Virtual Reality Augmented Reality Netw.*, 2017, pp. 13–18. [Online]. Available: http://doi.acm.org/10.1145/3097895.3097898

[4] L. Sun *et al.*, "Multi-path multi-tier 360-degree video streaming in 5G networks," in *Proc. 9th ACM Multimedia Syst. Conf. (MMSys)*, 2018, pp. 162–173. [Online]. Available: http://doi.acm.org/10.1145/3204949.3204978

[5] E. Kuzyakov. (2015). *Under the Hood: Building 360 Video*. [Online]. Available: https://code.facebook.com/posts/1638767863078802/under-the-hood-building-360-video/

[6] H.-C. Lin, C.-Y. Li, J.-L. Lin, S.-K. Chang, and C.-C. Ju, *An Efficient Compact Layout for Octahedron Format*, document JVET D0142, 2016.

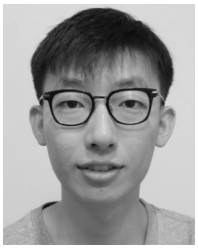
[7] E. Kuzyakov. (2016). *Next-Generation Video Encoding Techniques for 360 Video and VR*. [Online]. Available: https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/

[8] M. Zhou, *A Study on Compression Efficiency of Icosahedral Projection*, document JVET D0023, 2016.

[9] C. Zhang, Y. Lu, J. Li, and Z. Wen, *Segmented Sphere Projection (SSP) for 360-Degree Video Content*, document JVET D0030, 2016.

[10] G. V. der Auwera, H. M. Coban, and M. Karczewicz, *Truncated Square Pyramid Projection (TSP) For 360 Video*, document JVET D0071, 2016.

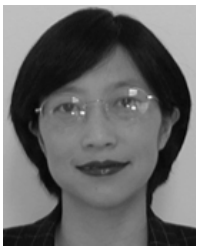
[11] F. Duanmu, Y. He, X. Xiu, P. Hanhart, Y. Ye, and Y. Wang, "Hybrid cubemap projection format for 360-degree video coding," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2018, p. 404.
[12] Y. He, X. Xiu, P. Hanhart, Y. Ye, F. Duanmu, and Y. Wang, "Content-adaptive 360-degree video coding using hybrid cubemap projection," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 1–6.
[13] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 583–586.
[14] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–7.
[15] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using scalable video coding," in *Proc. ACM Multimedia Conf.*, 2017, pp. 1689–1697. [Online]. Available: http://doi.acm.org/10.1145/3123266.3123414
[16] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski, "Optimal set of 360-degree videos for viewport-adaptive streaming," in *Proc. ACM Multimedia Conf.*, 2017, pp. 943–951. [Online]. Available: http://doi.acm.org/10.1145/3123266.3123372
[17] A. Zare, K. K. Sreedhar, V. K. M. Vadakital, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "HEVC-compliant viewport-adaptive streaming of stereoscopic panoramic video," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2016, pp. 1–5.
[18] Y.-K. Wang, Hendry, and M. Karczewicz, *Tile Based VR Video Encoding and Decoding Schemes*, document JCTVC X0077, 2016.
[19] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proc. 5th Workshop All Things Cellular, Oper., Appl. Challenges (ATC)*, 2016, pp. 1–6.
[20] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "HEVC-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proc. ACM Multimedia Conf.*, 2016, pp. 601–605.
[21] M. Graf, C. Timmerer, and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over HTTP: Design, implementation, and evaluation," in *Proc. 8th ACM Multimedia Syst. Conf.*, 2017, pp. 261–271
[22] S. Petrangeli, F. De Turck, V. Swaminathan, and M. Hosseini, "Improving virtual reality streaming using HTTP/2," in *Proc. 8th ACM Multimedia Syst. Conf.*, 2017, pp. 225–228
[23] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-aware adaptive 360 video streaming using tiles for virtual reality," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2174–2178.
[24] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge, and T. Schierl, "HEVC tile based streaming to head mounted displays," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 613–615.
[25] J. Son, D. Jang, and E.-S. Ryu, "Implementing motion-constrained tile and viewport extraction for VR streaming," in *Proc. 28th ACM SIGMM Workshop Netw. Oper. Syst. Support Digital Audio Video*, 2018, pp. 61–66. [Online]. Available: http://doi.acm.org/10.1145/3210445.3210455
[26] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "OpTile: Toward optimal tiling in 360-degree video streaming," in *Proc. ACM Multimedia Conf.*, 2017, pp. 708–716. [Online]. Available: http://doi.acm.org/10.1145/3123266.3123339
[27] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360ProbDASH: Improving QoE of 360 video streaming using tile-based HTTP adaptive streaming," in *Proc. ACM Multimedia Conf.*, 2017, pp. 315–323 http://doi.acm.org/10.1145/3123266.3123291
[28] J. M. Boyce, Y. Yan, J. Chen, and A. K. Ramasubramonian, "Overview of SHVC: Scalable extensions of the high efficiency video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 20–34, Jan. 2016.
[29] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol.*, 2012, pp. 109–120
[30] Y. Zhu, G. Zhai, and X. Min, "The prediction of head and eye movement for 360 degree images," *Signal Process., Image Commun.*, vol. 69, pp. 15–25, Nov. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596518304946
[31] M. A. Reina, K. McGuinness, X. Giró-i-Nieto, and N. E. O'Connor. (2017). "SaltiNet: Scan-path prediction on 360 degree images using saliency volumes." [Online]. Available: http://arxiv.org/abs/1707.03123
[32] A. D. Aladagli, E. Ekmekcioglu, D. Jarnikov, and A. Kondoz, "Predicting head trajectories in 360Â° virtual reality videos," in *Proc. Int. Conf. 3D Immersion (IC3D)*, Dec. 2017, pp. 1–6.
[33] J. Ling, K. Zhang, Y. Zhang, D. Yang, and Z. Chen, "A saliency prediction model on 360 degree images using color dictionary based sparse representation," *Signal Process., Image Commun.*, vol. 69, pp. 60–68, Nov. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596518302418
[34] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in VR spherical video streaming," in *Proc. 8th ACM Multimedia Syst. Conf. (MMSys)*, 2017, pp. 193–198. [Online]. Available: http://doi.acm.org/10.1145/3083187.3083210
[35] K. McCann, B. Bross, W. Han, I. K. Kim, K. Sugimoto, and G. J. Sulliva, *High Efficiency Video Coding (HEVC) Test Model 14 (HM 14) Encoder Description*, document N14970, 2014.
[36] E. Alshina, J. M. Boyce, A. Abbas, and Y. Ye, *JVET Common Test Conditions and Evaluation Procedures for 360Â° Video*, document JVET-G1030, 2017.



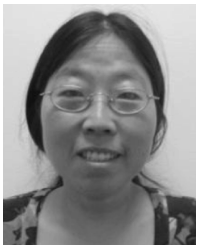
Liyang Sun received the bachelor's degree in optical and electronic information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2013, and the master's degree in electrical and computer engineering from Tandon School of Engineering, New York University, NY, USA, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include 360 degree video streaming, adaptive video streaming, and multipath technologies.

Fanyi Duanmu received the B.S. from the Beijing Institute of Technology, China, in 2009, the M.S. degree from the Polytechnic Institute of New York University, Brooklyn, NY, USA, in 2011, and the Ph.D. degree in electrical and computer engineering from Tandon School of Engineering, New York University, Brooklyn. His current research interests include video compression and delivery, screen content coding, 360-degree video processing and streaming, computer vision, and machine learning.

Yong Liu (F'17) received the bachelor's and master's degrees in automatic control from the University of Science and Technology of China in 1997 and 1994, respectively, and the Ph.D. degree from the Electrical and Computer Engineering Department, the University of Massachusetts Amherst, Amherst, in 2002. He joined Tandon School of Engineering, New York University as an Assistant Professor in 2005, where he is currently an Associate Professor with the Electrical and Computer Engineering Department. His general research interests lie in modeling, design, and analysis of communication networks. His current research interests include multimedia networking, network measurement, online social networks, and recommender systems. He was a recipient of the IEEE Communications Society Best Paper Award in Multimedia Communications in 2008, the IEEE Conference on Computer and Communications Best Paper Award in 2009, and ACM/USENIX Internet Measurement Conference Best Paper Award in 2012.



Yao Wang (F'04) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California at Santa Barbara in 1990. Since 1990, she has been on the Faculty of Electrical and Computer Engineering, Tandon School of Engineering, New York University (formerly Polytechnic University), Brooklyn, NY, USA. She has authored a textbook *Video Processing and Communications*, and has published more than 250 papers in journals and conference proceedings. Her current research areas include video communications, multimedia signal processing, and medical imaging. She was an elected Fellow of the IEEE in 2004 for his contributions to video processing and communications. She was a recipient of the New York City Mayor's Award for Excellence in Science and Technology in the Young Investigator Category in 2000. She was a co-recipient of the IEEE Communications Society Leonard G. Abraham Prize Paper Award in communications systems in 2004, and also a co-recipient of the IEEE Communications Society Multimedia Communication Technical Committee Best Paper Award in 2011. She has served as an Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. She was a Keynote Speaker at the 2010 International Packet Video Workshop.



Yinghua Ye received the Ph.D. degree in electrical engineer from the City University of New York in 2000. From 2000 to 2016, she was with Nokia Bell Labs as a Senior Researcher. She joined Futurewei technologies as a Senior Staff Engineer in 2016. She has published more than 50 technical papers and filed more than 30 patent applications of which eight patents have been granted. She had made several Standard contributions to IEEE802.3ah, and UPnP forum. Her work has covered many topics, such as intelligent transportation system, VR streaming in SDN in mobile networks, mobile network virtualization, mobile video streaming optimization, traffic offloading in mobile network, mobile IP, LTE and CDMA interworking, WiMAX, optical networks and EPON, and UPnP.

Hang Shi received the master's degree in computer science from the University of Southern California in 2000. From 2000 to 2015, he was with Cisco Systems as a Technical Leader. He joined Futurewei technologies in 2015, where he is currently a Principal Engineer. He has published more than 15 technical papers and filed more than 15 patent applications. His work has covered many areas, such as distributed computing, routing protocols, quality of services, SDN, network function virtualization, voice over IP, and data center networking.

David Dai received the M.Sc. degree from the University of Missouri–Columbia. He is currently the Senior Technical Director of Futurewei Technologies. He was one of the founding engineering team and architect at Embrane (acquired by Cisco), a network virtualization and SDN pioneer. He had also been with Andiamo (acquired by Cisco), where he held various technical leadership and managerial positions in data center infrastructure, CDN, Videoscape, and cloud computing. His past activities in Silicon Valley include product development in the area of networking technologies, ranging from frame relay and ISDN router, DSLAM, fixed wireless to Ethernet switch, optical switch, and SAN switch. His interests are focused on enabling ICT infrastructures, ranging across mobile edge computing, service based architecture, and cloud-based networking and service delivery.