Robust Multi-User 360° Virtual Reality Video Delivery

Jean Abou Rahal, and Gustavo de Veciana
The University of Texas at Austin, USA
Email: {jeanabourahal, deveciana}@utexas.edu

***Abstract*—We propose a jointly optimized edge-based framework to multi-user 360° Virtual Reality (VR) stored video streaming. At its core is a predictor that estimates a statistical model for users' future sequences of viewing directions (viewports) based on a Deep Recurrent Neural Network (DRNN). Such predictions, along with the tracking of users' past quality of experience, are used in a novel scheduler that fairly prioritizes the delivery of the high definition content in viewing directions which are most likely to be relevant. We achieve robustness to view prediction errors by ensuring low definition content for all viewports is always delivered. We achieve robustness to wireless capacity variations by proactively delivering (possibly through multicasting) and caching high definition content when the available capacity to users is high.**

I. Introduction

The ability that Virtual Reality (VR) systems have to deliver immersive experiences to customers is increasingly becoming attractive and has shifted high-end VR manufacturers from designing wired head-mounted devices (HMDs) to enabling fully wireless solutions, which have leveraged the advantages of edge servers to meeting VR systems' demands [1], [2]. Such approaches have unlocked the advantages of wireless technologies with high-throughput in providing timely delivery of massive amounts of data to users' VR devices and have paved the way towards more interactive VR experiences among multiple users. But with such advancements come major challenges. Providing customers with such VR experiences is burdensome from multiple perspectives. For instance, the variability in users' wireless channels may lead to disappointment in the overall users' experience by limiting the transfer of data streams, e.g., users traveling in a car/train watching VR videos on their headsets may be more prone to Quality-of-Experience (QoE) disruptions. Further, high-quality 360° VR videos require massive amounts of data to be constantly streamed, which may overwhelm shared communication resources and result in poor user experience. Motivated by the fact that users only view a portion of the 360° video content, viewport-adaptive streaming solutions that only stream in high quality what the users are viewing or likely to view have emerged as a key technique to efficiently serve users and save on communication resources [3]–[7].

With that in mind, we propose a new approach to users' viewing predictions, which we refer to as a statistical model for the users' viewing processes in 360° VR video applications that not only predicts the portion of the video content that's most likely to be viewed in the future, but predicts as well a set of possible viewports over a future time-window along with their probabilities of being viewed by the users. The advantage of such a model is that it captures in a more detailed manner the level of certainty regarding the users' viewport predictions which may support more robust decision-making in deciding how to serve users. General settings may include users with high and/or low uncertainty regarding their predicted viewing patterns. Users with high prediction uncertainty may burden the communication medium due to the huge amount of data sent which may still result in poor users' QoE due to a small hit rate of their viewports. On the other hand, users with low uncertainty may have overall better VR experiences due to the small amount of data required to achieve such good VR performance. The question of VR QoE fairness arises in such settings where users with different levels of uncertainty regarding their predictions and experiencing potentially different network conditions share the same resources. Such challenges encountered in fully immersive VR systems need to be tackled through careful design of metrics and policies that account for the variability in the network channel and prioritizes QoE fairness while servicing multiple users sharing the same resources.

II. System model

We consider a setting where a catalog of 360° VR videos are stored at a edge node co-located with a wireless Base Station (BS). Users equipped with mobile VR headsets initiate viewing sessions at random times by selecting the 360° VR video they wish to watch. For simplicity we assume that users are served by the same fixed edge/BS but more generally mobile users, e.g., within a car/bus, would be handed off to other edge/BS resources when appropriate.

All 360° VR videos at the edge have a frame rate of $\frac{1}{\beta}$ frames per second, i.e., the time between two successive video frames is β seconds. The user's mobile VR device tracks its 3DoF pose and reports it to the edge every β seconds as well. We shall consider a time-slotted system where each slot corresponds to a video frame which is indexed by τ and is of length β seconds. Users initiate 360° VR video viewing sessions at possibly different times and leave the network when the VR video they are watching ends. We let a^u be the time at which user u starts watching its selected video and we let w^u be the length of user u's viewing session in terms of the number of slots. We refer to users watching a 360° VR video

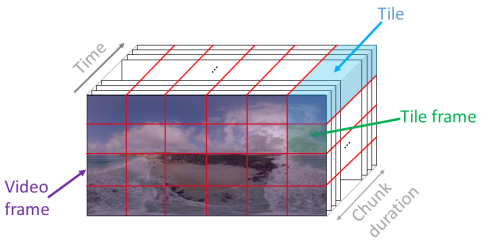


Fig. 1: Video chunk, tiles and tile frames.

on their VR headsets as active users. We let U_τ denote the set of active users at the start of slot τ.

Remark 1. *For simplicity, we assume that new users can become active only at the beginning of a slot τ.*

A. User's viewport and video compression model

A user's viewport $\boldsymbol{v}$ in a 360° VR video frame is a 3-D vector with its components being the pitch, roll and yaw angles respectively, which corresponds to the direction in which the user is looking at. We let $\mathcal{V}$ denote the set of all viewports/viewing directions that any user can look at. The viewport viewed by active user u in slot τ is denoted by $\boldsymbol{v}_\tau^{u,*}$.

Remark 2. *In the sequel we refer to a user's viewport as the direction in which the user is, or may be, looking at.*

Each 360° VR video is temporally segmented into video chunks in a manner which is compliant with the DASH paradigm [8], and each video chunk is spatially divided into tiles. Temporal video segmentation facilitates temporal compression by exploiting redundancy and allows for the efficient delivery of relevant content [9]. Each tile has the same duration and same number of frames as a video chunk. An example of a video frame, video chunk, its constituent tiles and tile frames is illustrated in Figure 1.

VR video encoding is performed using a layered approach [10], [11], where the availability of the lower layer tile encoding results in Low-Definition (LD) content and the *additional* availability of the higher layer tile encoding results in High-Definition (HD) content. Both low/high encodings of all tiles are available at the edge.

We let $\mathcal{F}_\tau^u$ denote the set of tile frames in the video frame to be displayed to user u at time τ. We let $h^u(\boldsymbol{v},\tau)$ be the function that maps viewport $\boldsymbol{v}\in\mathcal{V}$ that is viewed, or may be viewed by user u in slot τ to a subset of tile frames in $\mathcal{F}_\tau^u$. We note that a user's viewport maps to different sets of tile frames in different video frames of the same video. We let d_f^u for $f\in\mathcal{F}_\tau^u$, be the time at which tile frame f may be viewed by user u and we refer to this as the *viewing deadline* of f. Hence, for $f\in\mathcal{F}_\tau^u$, $d_f^u=\tau$. We further let $\mathcal{T}_\tau^u$ denote the set of tiles that belong to the video chunk that u is watching at time τ. We let $g(\cdot)$ be a one-to-one mapping function that maps a tile frame to the tile it belongs to. We finally let b_t denote the size in bits of HD tile $t\in\mathcal{T}_\tau^u$.

B. VR video delivery and user's Quality of Experience model

The edge/BS coordinates the delivery of VR video tiles in each slot τ, i.e., in between the time to display two successive video frames to active users U_τ. The edge/BS schedules the transmissions at the granularity of LD and HD tiles rather than scheduling LD and HD tile frames or video chunks. Every slot τ the edge server prioritizes the delivery of the LD tiles which are sent sequentially using a traditional video streaming protocol, e.g., DASH [8]. The edge guarantees the transmission of all LD tiles that belong to the next video chunk of each active user, motivated by the small size of the LD tiles and to provide robustness to fast changes in users' viewports. The edge/BS then predicts the set of viewports that users may view in future slots, maps them to the corresponding set of tiles and finally opportunistically schedules the transmission of a subset of the HD tiles to the users depending on the available capacity at the BS. HD tiles are sent to users to further enhance their VR experience. The delivery of HD tiles is supported by low-latency transport protocol such as QUIC [12], which can be utilized for streaming video content. We assume that users' headset devices have two buffers, one that caches LD tiles and another that caches HD tiles. For the remainder of this work, we will solely focus on the scheduling of HD tiles' transmissions from the edge/BS to the users, as only HD tiles cached in a user's device's buffer will account for the QoE.

We let $Q_\tau^{u,\pi}$ denote the set of HD tiles cached in user u's device buffer at time τ under HD tile scheduling policy π, and assume that u's buffer can cache in slot τ at most l_τ^u bits tied only to HD tiles. We hence propose to model a user's QoE in a VR session as a function of the set of HD tiles cached in its device's buffer and available prior to their viewing deadlines.

Definition 1. (User's QoE) *A user's QoE is defined as a function of the fraction of HD tile frames whose tiles are cached in the user's device's buffer at their deadline and that fall in the viewport viewed by the user. User u's QoE under an HD tile scheduling policy π is defined as,*

$$\mathrm{QoE}^{u,\pi}=\frac{1}{w^u}\sum_{\tau=a^u}^{a^u+w^u} q\left(\frac{1}{|h^u(\boldsymbol{v}_\tau^{u,*},\tau)|}\sum_{f\in h^u(\boldsymbol{v}_\tau^{u,*},\tau)}\mathbb{1}_{\{g(f)\in Q_\tau^{u,\pi}\}}\right),$$

where $\mathbb{1}_{\{g(f)\in Q_\tau^{u,\pi}\}}$ takes value 1 if HD tile $g(f)$ is in $Q_\tau^{u,\pi}$ and 0 otherwise, $|h^u(\boldsymbol{v}_\tau^{u,*},\tau)|$ is the number of tile frames in $\boldsymbol{v}_\tau^{u,*}$ and where $q(\cdot)$ is a strictly increasing concave function which provides some flexibility in modeling a user's QoE in 360° VR applications.

So, a user's QoE marginal gain for a time slot may decrease as the instantaneous hit rate of tiles cached in the user's buffer increases. This is captured through the $q(\cdot)$ function.

We now propose a multi-user system utility function that realizes QoE trade-offs across users and defined as follows.

Definition 2. (Multi-user system utility) *The multi-user system utility is defined as the sum of the natural logarithmic*

function of the users' QoEs. The utility function under an HD tile scheduling policy π is defined as

$$v(\mathbf{QoE}^{\pi}) = \sum_{u \in U} \log\left(\mathrm{QoE}^{u,\pi}\right),$$

where $\mathbf{QoE}^{\pi} := (\mathrm{QoE}^{u,\pi} : u \in U)$ *and* $U = \bigcup_{\tau} U_{\tau}$.

The natural logarithmic function encourages a degree of QoE fairness across users.

C. Edge/BS resources

The BS provides both unicast and multicast transmission services. Its resources may be shared with other high priority traffic as well and thus possibly only limited congested wireless resources are available to transmit HD tiles. We assume that the edge has a rough estimate of the available wireless resources to transmit HD tiles in slot τ once LD tiles and other high priority traffic are delivered. We let θ_{τ}^{u} denote the estimate of the maximal number of bits that the edge can transmit to user u throughout slot τ if all available network resources are allocated to u. We further let $\rho_{\tau} \leq 1$ be the fraction of the available resources that can be used for multicast transmissions. The edge/BS finally proactively schedules the transmission of a subset of HD tiles to their corresponding users based on its estimate of the available resources, and receives feedback regarding their successful delivery. The edge guarantees that no tile is delivered more than once to the same user. If an HD tile t is scheduled in slot τ for multicast transmission to a subset of active users $A_{\tau} \subseteq U_{\tau}$, the edge/BS estimates the maximal number of bits that it can transmit to all users in A_{τ}. For simplicity, we shall assume that the edge/BS adapts the transmission rate to match that of the user with the worst transmission rate in A_{τ}. Hence, the proportion of resources reserved for the multicast transmission of t to users in A_{τ} in slot τ is $\max_{u \in A_{\tau}} \frac{b_t}{\theta_{\tau}^{u}}$.

Finally at the beginning of slot $\tau+1$, the edge discards any HD tiles that were not successfully transmitted in slot τ to their corresponding users. Note that tiles cached in the user's buffer and that belong to a specific video chunk are deleted once frames therein are done being viewed by the user.

III. Multi-user system utility maximization

We define the following notation to properly account for unicast and multicast transmissions at the BS.

- $\mu_{\tau}^{\pi,u} \leq \theta_{\tau}^{u}$, is the unicast transmission rate assigned to user $u \in U_{\tau}$ under policy π in slot τ.
- B_{τ}^{π} is the set of HD tiles that are multicast to subsets of users in U_{τ} in slot τ under π.
- $M_{\tau,t}^{\pi} \subseteq U_{\tau}$ is the subset of users to which HD tile $t \in B_{\tau}^{\pi}$ is multicast to in slot τ under π.
- $R_{\tau,t}^{\pi} \subseteq U_{\tau}$ is the subset of users to which HD tile t is unicast to in slot τ under π.

We now formulate the multi-user system utility maximization problem.

Problem 1. (Multi-user system utility maximization).

$$\max_{\pi \in \Pi} v(\mathbf{QoE}^{\pi})$$

$$\text{s.t.} \sum_{t \in B_{\tau}^{\pi}} \max_{u \in M_{\tau,t}^{\pi}} \frac{b_t}{\theta_{\tau}^{u}} \leq \rho_{\tau}, \ \forall \tau, \tag{1}$$

$$\sum_{u \in U_{\tau}} \frac{\mu_{\tau}^{u,\pi}}{\theta_{\tau}^{u}} \leq 1 - \sum_{t \in B_{\tau}^{\pi}} \max_{u \in M_{\tau,t}^{\pi}} \frac{b_t}{\theta_{\tau}^{u}}, \ \forall \tau, \tag{2}$$

$$\sum_{t \in Q_{\tau}^{u,\pi}} b_t \leq l_{\tau}^{u}, \forall u \in U_{\tau}, \forall \tau, \tag{3}$$

where Π *denotes the set of all policies that schedule the transmission of HD tiles to users.*

Equation (1) enforces a constraint on the proportion of available resources that can be used for multicasting, while Equation (2) is a constraint on the fraction of time that a user's channel can be reserved for unicasting. Equation (3) enforces a constraint on the number of HD tiles that can be cached in a user's buffer.

IV. Proposed algorithm for Problem 1

We now propose an algorithm geared at maximizing the multi-user system utility in Problem 1. It comprises of three steps repeated every slot τ as follows. First, the edge predicts the set of viewports that the user may view in future slots through a predictor which we refer to as a Statistical Model Predictor (*SMP*) formally described in Subsection IV-A, then maps the tile frames in future predicted viewports to their corresponding sets of HD tiles (Subsection IV-B). Second, the edge/BS predicts the communication resources available in slot τ for the transmission of predicted HD tiles. For simplicity, we will assume that a crude estimate of the available resources is available at the edge/BS every slot τ. Third, the edge/BS uses its prediction of the relevant tiles and available capacity to schedule the HD tiles to be either unicast or multicast. We refer to our policy as π^{F} as it is geared at achieving a fair allocation of QoE across users. π^{F} is described in detail in Subsection IV-C. We refer to our proposed algorithm as *SMP*+π^{F}.

A. Statistical model for a user's viewing process

We propose to predict a statistical model for a user's future viewing process given the past viewing history. The model corresponds to a tree of depth ω and branching factor of at most α, where each tree node corresponds to a possible viewport seen at a future time slot – to reduce the model's complexity and recognizing that viewports may shift relatively slowly as compared to a frame duration, the tree's nodes at different depths correspond to viewports sampled every γ slots. Figure 2 exhibits an estimated statistical model for user u's viewing process with $\alpha = 2, \omega = 2$ and $\gamma = 5$ given the viewing history up to and including slot τ.

We shall let $\hat{\mathcal{V}}_{\tau}^{u} := \{n_{\tau,i}^{u} : i \in \{1, \ldots, m_{\tau}^{u}\}\}$ denote the set of nodes in the tree for user u's statistical model as estimated at time τ whence for future slots $\tau+\gamma, \ldots, \tau+\omega\gamma$, which comprises of m_{τ}^{u} nodes, and where $n_{\tau,i}^{u} = (\hat{\boldsymbol{v}}_{\tau,i}^{u}, k_{\tau,i}^{u}, \hat{p}_{\tau,i}^{u})$ is a node indexed by i and characterized by (1) a predicted viewport $\hat{\boldsymbol{v}}_{\tau,i}^{u} \in \mathcal{V}$; (2) a slot $k_{\tau,i}^{u} \in \{\tau+\gamma, \ldots, \tau+\omega\gamma\}$; and

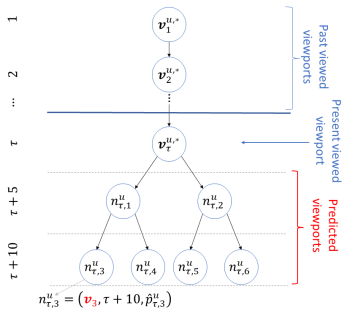


Fig. 2: Past and present viewed viewports and the estimated statistical model for user u's viewing processes at time $\tau = m$ with $\alpha = 2, \omega = 2$ and $\gamma = 5$.

(3) an estimated probability $\hat{p}^u_{\tau,i}$ given the viewing process history from $\tau - \delta + 1$ up to τ for $\delta > 1$. We note that two nodes in the same tree level may have the same characteristics, but different viewing trajectories.

We propose a predictor which we refer to as the Statistical Model Predictor (*SMP*) that any given slot say τ takes as input the sequence of δ most recently viewed viewports, i.e., $\boldsymbol{V}^{u,*}_{[\tau-\delta+1,\tau]} = (\boldsymbol{v}^{u,*}_{\tau-\delta+1}, \boldsymbol{v}^{u,*}_{\tau-\delta+2}, \ldots, \boldsymbol{v}^{u,*}_{\tau})$, and outputs a statistical model for the user's viewing process for slots $\tau+\gamma, \ldots, \tau+\omega\gamma$. Note that the overall prediction window is $\omega\gamma$. Ideally to obtain accurate predictions of the future, ω should be large and γ should be small. The *SMP*'s main block is a Deep Recurrent Neural Network (DRNN) that takes as input a sequence of δ viewports up to the most recent slot τ and estimates the probability of viewing each possible viewport in slot $\tau+\gamma$.

The statistical model is predicted through recursive executions of the DRNN. In slot τ, the first execution of the DRNN estimates the viewports' probabilities of what is viewed in slot $\tau+\gamma$. Next, each of the α viewports with the highest estimated probabilities of being viewed in slot $\tau+\gamma$ is concatenated to $\boldsymbol{V}^{u,*}_{[\tau-\delta+2,\tau]}$ to form α new sequences of viewports. Each newly formed sequence of viewports is then fed to the DRNN to estimate the statistical model for the viewing process in slot $\tau+2\gamma$. This process is repeated over ω time slots delivering the tree model for the user's viewing process. Details regarding the architecture of the DRNN and *SMP* are left out of the paper for space constraints. The number of recursive DRNN executions the *SMP* requires to estimate a statistical model of depth ω and branching factor α is thus at most $\frac{\alpha^\omega - 1}{\alpha - 1}$. This structure permits one to increase the value of γ while reducing ω so as to increase the depth of the future predictions while reducing the complexity of the *SMP*.

Recall that a viewport at a given time maps to a set of tile frames. We let $\hat{\mathcal{F}}^u_\tau$ denote the set of tile frames corresponding to the tuples, i.e., viewports at given times, in $\hat{\mathcal{V}}^u_\tau$. Further we let $\hat{\mathcal{T}}^u_\tau$ denote the set of tiles that tile frames in $\hat{\mathcal{F}}^u_\tau$ map to.

B. Weights and deadlines of predicted HD tile frames and tiles

Given a constraint on the resources available for the transmission of HD tiles, the edge can schedule only a subset of the predicted tiles and so the most relevant HD tiles will need to be selected and scheduled. We hence propose to associate a weight with each predicted tile frame based on the probabilities associated with predicted viewports that map to this tile frame, in an approach that would simply capture its priority/importance in comparison with other tile frames. Note that a tile frame may fall in more than one viewport predicted to be viewed by a user in a particular video frame and hence the weight of a tile frame is defined as follows.

Definition 3. (Weight of a predicted HD tile frame) *The weight of a predicted HD tile frame $f \in \hat{\mathcal{F}}^u_\tau$ is defined as follows,*

$$\eta^u_{\tau,f} = \sum_{i: n^u_{\tau,i} \in \hat{\mathcal{V}}^u_\tau} \frac{p^u_{\tau,i}}{|h^u(\hat{\boldsymbol{v}}^u_{\tau,i}, k^u_{\tau,i})|} \mathbb{1}_{\{h^u(\hat{\boldsymbol{v}}^u_{\tau,i}, k^u_{\tau,i})\}}.$$

Note that not all viewports will map to the same number of tile frames in a specific video frame [13].

Remark 3. *In general, the viewport's probability of being viewed may be distributed non-homogeneously over the tile frames that belong to the viewport, based, for example, on their proximity to the center of the viewport. In Definition 3, for simplicity we assume that the viewport's probability of being viewed is divided uniformly over all such tile frames.*

In turn, the weight of a tile is formally defined below.

Definition 4. (Weight of a predicted HD tile) *The weight of an HD tile $t \in \hat{\mathcal{T}}^u_\tau$ is the sum of the weights of all HD tile frames in $\hat{\mathcal{F}}^u_\tau$ that map to tile t and defined as follows,*

$$\lambda^u_{\tau,t} = \sum_{f \in \hat{\mathcal{F}}^u_\tau} \eta^u_{\tau,f} \mathbb{1}_{\{g(f)=t\}}.$$

The viewing deadlines of HD tile frames can be used to define deadlines for HD tiles. The *viewing deadline* of a predicted HD tile is defined below.

Definition 5. (Viewing deadline of a predicted HD tile) *The viewing deadline of an HD tile $t \in \hat{\mathcal{T}}^u_\tau$ is tied to the earliest tile frame deadline among all tile frames in $\hat{\mathcal{F}}^u_\tau$ that map to t and is given as,*

$$e^u_{\tau,t} = \min_{f \in \hat{\mathcal{F}}^u_\tau : g(f)=t} d^u_f.$$

C. Our proposed HD tile scheduling policy π^F

Every slot τ, π^F schedules a subset of predicted HD tiles for either unicast or multicast transmissions based on the estimate of the available resources at the BS, namely $\boldsymbol{\theta}_\tau := (\theta^u_\tau : u \in U_\tau)$ and ρ_τ, with the intent of maximizing the multi-user QoE utility while achieving QoE fairness among users.

π^F keeps track of an estimate of the experienced QoE up to time τ for every user $u \in U_\tau$ denoted r^{u,π^F}_τ. Every τ, π^F updates r^{u,π^F}_τ for all $u \in U_\tau$ as a weighted average between $r^{u,\pi^F}_{\tau-1}$ and the concave function $q(\cdot)$ of the fraction of HD tile frames cached in u's device's buffer that fall in the viewport viewed by u in slot τ.

π^F next selects an HD tile t^* and the user u^* (described later) it has to be sent to, and then determines whether t^* can be multicast to u^* and other active users in the set $M^{\pi^F}_{\tau,t^*}$ if Constraints 1, 2 and 3 in Problem 1 are satisfied. Otherwise, π^F unicasts tile t^* to user u^* if Constraints 2 and 3 in Problem 1 are satisfied. Else, tile t^* is not scheduled for transmission to u^*. A queue $L^{\pi^F}_\tau$ keeps track of the order in which selected tiles will be scheduled for transmission to their corresponding users which is tied to the order in which they were selected.

We let $\hat{c}^u_\tau(\cdot)$ be the function that evaluates user u's expected future cumulative QoE over sampled slots $\tau+\gamma,\ldots,\tau+\omega\gamma$, given the set of HD tiles cached in Q^{u,π^F}_τ and let X^u_τ be a random variable that denotes the viewport viewed by user u in slot τ.

$\hat{c}^u_\tau(Q^{u,\pi^F}_\tau)$ is provided in Eq.(5), where $\hat{\mathbb{E}}_\tau[\cdot]$ denotes the expectation taken with respect to the predicted process over sampled times and given the set of HD tiles cached in Q^{u,π^F}_τ, where $h^u(X^u_{\tau+\alpha\gamma},\tau+\alpha\gamma)$ is the set of tile frames that fall in viewport $X^u_{\tau+\alpha\gamma}$ in slot $\tau+\alpha\gamma$, and where $g(f)$ maps tile frame f to its corresponding tile.

The marginal increase in the expected future cumulative QoE of user u over sampled slots after scheduling HD tile t to user u and given the set of HD tiles cached in u's device's buffer is denoted as,

$$\hat{\nabla}^{u,\pi^F}_{\tau,t}(Q^{u,\pi^F}_\tau)=\hat{c}^u_\tau(Q^{u,\pi^F}_\tau\bigcup\{t\})-\hat{c}^u_\tau(Q^{u,\pi^F}_\tau). \quad (4)$$

For simplicity, we refer to $\hat{\nabla}^{u,\pi^F}_{\tau,t}(Q^{u,\pi^F}_\tau)$ as $\hat{\nabla}^{u,\pi^F}_{\tau,t}$. The derivation of an approximation of $\hat{\nabla}^{u,\pi^F}_{\tau,t}$ is left out of the paper due to space constraints. We point out that if $q(\cdot)$ is a linear function such that $q(x)=x$, then $\hat{\nabla}^{u,\pi^F}_{\tau,t}=\lambda^u_{\tau,t}$, i.e., it is equivalent to the weight of HD tile t evaluated in slot τ.

π^F greedily selects the tile t^* and the user u^* that maximizes $\hat{\nabla}^{u^*,\pi^F}_{\tau,t^*}/\hat{r}^{u^*,\pi^F}_{e^{u^*}_{\tau,t^*}-1}$, where $\hat{r}^{u^*,\pi^F}_{e^{u^*}_{\tau,t^*}}$ is the estimated QoE of user u^* at the deadline of t^*, i.e., at $e^{u^*}_{\tau,t^*}$, which is geared at achieving a fair allocation of QoE across users. $\hat{r}^{u^*,\pi^F}_{e^{u^*}_{\tau,t^*}}$ is evaluated based on $\hat{r}^{u,\pi^F}_\tau=r^{u,\pi^F}_\tau$ as well as on both the set of HD tile frames whose tiles are cached in u's device buffer in slot τ and on their weights that account for their probability of being viewed.

V. Simulations

We conducted simulation based evaluations to explore the performance of *SMP*+π^F with respect to other baseline policies which we introduce in Subsection V-C. We assume in our simulations that all LD tiles are successfully delivered to the active users' headsets prior to their deadlines and only focus on scheduling HD tiles. We further assume that the scheduled HD tiles under any policy are successfully transmitted, i.e., an accurate prediction of network resources.

A. Dataset Preparation

We consider the dataset made available by [14] which provides the 3DoF poses of 50 different users tracked while they were watching HD 360° VR videos from a catalog of 10 videos. The videos are encoded at 30 fps and are 60 seconds long. The equirectangular (EQR) projection of each of the video frames is divided into 200 square tiles of 192x192 pixels. The dataset also provides the ground-truth mapping of the users' 3DoF poses in each video frame to the corresponding set of observed tile frames. For each user and each 360° VR video, there is a sequence of 1800 3DoF poses and the corresponding tile frames they map to. We consider video chunks of 1 second duration each, i.e., each tile comprises of 30 tile frames. We consider 35 users from the dataset for training.

B. SMP parameters

We set $\alpha=2,\gamma=5$ and $\omega=2$ and 4.

C. Baseline algorithms

In addition to *SMP*+π^F we consider three other baseline algorithms, namely *ML*+π^G, *SMP*+π^G and *SMP*+π^{RR}. All baseline policies differ from *SMP*+π^F in the HD tile and user selection, i.e., the way they select t^* and u^* but otherwise proceed similarly to *SMP*+π^F. *ML*+π^G additionally differs from *SMP*+π^F in the viewports' prediction methodology.

- *ML*+π^G: Determines among the predicted viewports which is most-likely to be viewed in slots $\tau+\gamma,\ldots,\tau+\omega\gamma$ and uses a greedy scheduling policy, i.e., prioritizes scheduling (1) tiles with earliest deadlines and (2) tiles with the largest weights. Every slot τ, the sequence of viewports most-likely to be viewed in slots $\tau+\gamma,\ldots,\tau+\omega\gamma$ is predicted for all active users.
- *SMP*+π^G: Estimates a user's statistical model for its viewing process and selects t^* and u^* under policy π^G.
- *SMP*+π^{RR}: Estimates a user's statistical model for its viewing process and selects active users in a round-robin-like fashion then selects for each selected user u^* the HD tile t^* with the largest weight.

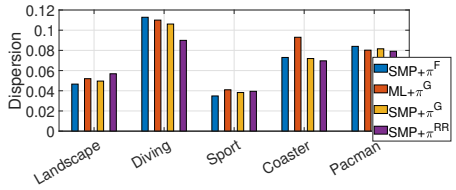
D. Network conditions

In our simulation, we will use the traces for users' time-varying capacity provided in [15]. These provide millisecond level wireless capacity accuracy. We use the file named "*Verizon-LTE-driving.down*" which provides traces for around 1400 seconds, and evaluate the maximal transmission rate for every slot τ, i.e., over 33 milliseconds time windows.

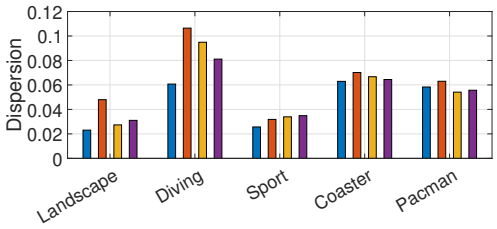
E. Model

We consider a setting where 4 randomly selected users from the testing dataset $u_1,\ldots,u_4$, start watching 360° VR videos at the same time and for a duration of 30 seconds. We thus

$$\hat{c}^u_\tau(Q^{u,\pi^F}_\tau)=\sum_{\alpha=1}^{\omega}\hat{\mathbb{E}}_\tau\left[q\left(\frac{1}{|h^u(X^u_{\tau+\alpha\gamma},\tau+\alpha\gamma)|}\sum_{f\in h^u(X^u_{\tau+\alpha\gamma},\tau+\alpha\gamma)}\mathbb{1}_{\{g(f)\in Q^{u,\pi^F}_\tau\}}\right)\right]. \quad (5)$$



(a) Users' QoE dispersion for $\omega = 2, \gamma = 5$



(b) Users' QoE dispersion for $\omega = 4, \gamma = 5$

Fig. 3: Users' QoE dispersion for $\omega = 2, 4$ and fixed $\gamma = 5$, when users are watching the same 360° VR video and experience the same network capacity.

simulate our results for the first 30 seconds of any VR video that the users are watching. We assume that each HD tile has a size of 0.1 Mbits and set $\epsilon = 10^{-3}$. We let $q(\cdot)$ be a linear function where $q(x) = x$. We provide bar plots for the users' QoE results, users' QoE dispersion which corresponds to the ratio between the QoE standard deviation and the average QoE achieved under the proposed policies. We observe throughout our simulations that the average number of HD tiles scheduled per slot is somewhat similar for all the proposed policies.

F. Users watching the same 360° VR video

We consider the setting where all active users are watching the same 360° VR video and we provide results for 5 different VR videos.

1) Homogeneous network capacity: We assume that the maximal number of bits available for the transmission of HD tiles is the same for all active users throughout their VR sessions, i.e., $\theta_\tau^{u_i} = \theta_\tau$ for all τ and for all $i \in \{1, \cdots, 4\}$. We further set $\rho_\tau = 0.6$ for all τ. We use the network capacity trace from time 800 to 830 seconds since it exhibits fluctuations and has a mean capacity of 0.833 Mbits per slot. The results are provided in Figure (3).

We observe that the users' QoE dispersion decreases as ω increases from 2 to 4 which confirms that proactively scheduling HD tiles whose tile frames are predicted to be viewed in future slots provides robustness to network capacity fluctuations. Further, for $\omega = 4$, *SMP+*π^F achieves an overall lower QoE dispersion as compared to *SMP+*π^G and *ML+*π^G. This observation has showcased that neither solely scheduling HD tiles whose tile frames belong to viewports along the most-likely viewport path nor greedily scheduling HD tiles by prioritizing predicted tiles with earliest deadlines have proven to achieve best performance. to maximize the average QoE while minimizing the standard deviation.

VI. Conclusion

In this paper, we proposed various 360° VR video delivery algorithms based on estimating statistical models for users' future viewing paths. This allows us to consider going beyond myopic approaches that focus only on the next most-likely viewport, and to build strategies that are robust to uncertainty in users' viewing processes, estimation errors and users' capacity variations.

VII. Acknowledgments

This work was supported by the National Science Foundation Award CNS- 2212202.

References

[1] C.-L. Fan, W.-C. Lo, Y.-T. Pai, and C.-H. Hsu, "A survey on 360 video streaming: Acquisition, transmission, and display," *Acm Computing Surveys (Csur)*, vol. 52, no. 4, pp. 1–36, 2019.

[2] L. Liu, R. Zhong, W. Zhang, Y. Liu, J. Zhang, L. Zhang, and M. Gruteser, "Cutting the cord: Designing a high-quality untethered vr system with low latency remote rendering," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 68–80.

[3] S. Zhang, M. Tao, and Z. Chen, "Exploiting caching and prediction to promote user experience for a real-time wireless vr service," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[4] B. Yu, X. Zhang, and I. You, "Collaborative cache allocation and transmission scheduling for multi-user in edge computing," *IEEE Access*, vol. 8, pp. 163 953–163 961, 2020.

[5] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Edge computing meets millimeter-wave enabled vr: Paving the way to cutting the cord," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[6] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communication, computing and caching for mobile vr delivery: Modeling and trade-off," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[7] M. Palash, V. Popescu, A. Sheoran, and S. Fahmy, "Robust 360° video streaming via non-linear sampling," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[8] ISO. (2019) Dynamic adaptive streaming over http (dash). [Online]. Available: https://www.iso.org/standard/75485.html

[9] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey," *Signal processing: Image communication*, vol. 16, no. 5, pp. 477–500, 2001.

[10] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using layered video coding," in *2017 IEEE Virtual Reality (VR)*. IEEE, 2017, pp. 347–348.

[11] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *IEEE Transactions on circuits and systems for video technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[12] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.

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

[14] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "360 video viewing dataset in head-mounted virtual reality," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 211–216.

[15] M. PROJECT, "Mahimahi wireless network traces," https://github.com/ravinet/mahimahi/tree/master/traces, 2019.