Multi-Connectivity and Edge Computing for Ultra-Low-Latency Lifelike Virtual Reality

Jacob Chakareski* and Sabyasachi Gupta†
*New Jersey Institute of Technology, †Southern Methodist University

***Abstract*—We explore a novel multi-user mobile VR system for streaming scalable 8K 360° video at high reliability and immersion fidelity, and low interactive latency, via a synergistic integration of scalable 360° tiling, dual-band millimeter wave (mmWave) and Wi-Fi transmission, and edge computing. High rate directed mmWave links are studied to send VR viewport-specific high-quality enhancement layers of the 360° content to the individual users, while Wi-Fi broadcast of the base layer of the entire 360° panorama is sent to all users, to augment the system's reliability. The viewport-specific enhancement layers can comprise compressed and raw 360° tiles, decoded first at the edge server. We explore the joint optimization of the mmWave access point to user association, the choice of 360° tiles to be transmitted decompressed, the allocation of mmWave data rate across the compressed tiles in a viewport-specific enhancement layer, and the allocation of computing resources at the edge server and user devices. Our objective is to maximize the minimum delivered VR immersion fidelity across all users, given transmission, latency, and computing constraints. We demonstrate that our framework can enable a significant improvement in immersion fidelity (8 dB to 10 dB) and spatial resolution (8K vs. 4K), over MPEG-DASH that uses Wi-Fi transmission only. We also show that an increasing number of raw 360° tiles are sent, as the mmWave link rate or the edge server/user computing power increase, exploring rigorously here the fundamental interplay between computing and communication capabilities, end-to-end system latency, and delivered VR immersion fidelity.**

***Index Terms*—Mobile virtual reality, scalable 360° video tiling, dual-band Wi-Fi/millimeter wave wireless streaming, edge computing, resource allocation, geometric programming, rate-computing-delay interplay.**

I. Introduction

A. Background

Virtual reality (VR) technologies are increasingly becoming popular. Related applications include entertainment and gaming, education and training, healthcare, advertising, and social media. It is expected that VR technology will represent a $120 billion market by 2022 [1]. 360° video is an integral part of VR systems and can enable remote scene immersion for a VR user experiencing it. Relative to traditional video streaming [2–10], VR based 360° video streaming introduces these challenging requirements: ultra high data rate, ultra low response latency, and intensive computing [11]. Thus, at present, only low-quality/low-resolution 360° videos can be streamed over wired networks. In the mobile setting, the quality of experience is even worse, due to the much lower wireless bandwidth and computing capability of such devices, attached to a VR headset. However, seamless untethered VR applications integrating high-fidelity real remote scene 360° content are expected to have the highest societal impact, advancing quality of life, the global economy, and energy conservation [12–14]. *Enabling such applications is our objective.*

The latency in 360° video streaming comprises communication/computing delays that need to be constrained to 10-20 milliseconds end-to-end [15]. Relative to traditional video, 360° video has much higher resolution and temporal frame rates, and thus requires much higher bandwidth to stream the remote scene it represents. A high quality 360° video at 120 frames per second and spatial resolution of 12K, as recommended by MPEG, can easily consume bandwidth of multiple Gigabits-per-second (Gbps) [16].

B. Proposed framework

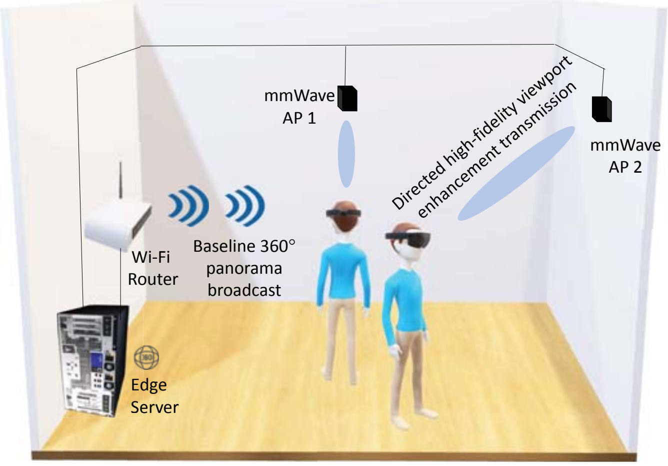

Fig. 1: Future mobile VR arcade with integrated dual-band scalable 360° streaming, edge computing, and millimeter wave capabilities.

We explore a novel streaming system for future untethered VR that enables high immersion fidelity and low interactive latency, illustrated in Figure 1. *It synergistically integrates for the first time* high fidelity scalable 8K 360° video, edge computing, and dual-band Wi-Fi and millimeter wave wireless transmission. By using high computing/high transmission rate capabilities at the network edge, our system enables decompressing select tiles of the 360° content at the edge server and transmitting them as raw data over mmWave links, aiming to minimize the overall system delay. Thereby, we reduce the delay induced by decompressing these tiles at the user device and enhance the system efficiency, at the same time. In our system, a user is assigned to a mmWave access point (AP) and viewport-specific enhancement layers of the scalable 360° content are sent to this user using mmWave transmission, comprising both raw and compressed tiles, as selected by a related optimization.

mmWave links are fragile due to their narrow-beam directed nature, and can be interrupted due to a transient line-of-sight loss or blockage. Similarly, predicting the user's imminent navigation actions, for proactive delivery of the respective portions of the 360° content, to maintain the application's ultra low latency nature, can be imperfect. Both of these phenomena can impact the delivered quality of experience. Hence, to address these challenges and augment the system's reliability and delivered immersion fidelity/application quality, we compensate for the prospective mmWave link uncertainty and VR viewport prediction error, by broadcasting the base layer of the entire 360° panorama to all users, over Wi-Fi.

This work was supported in part by NSF Awards CCF-1528030, ECCS-1711592, CNS-1836909, and CNS-1821875, and in part by research gifts and an Adobe Data Science Award from Adobe Systems.

C. Related work

Tiling-based streaming [17–20] is a popular approach for efficient resource utilization, where 360° video frames are spatially divided into smaller rectangular regions called tiles. Viewport-adaptive rate-distortion optimized tiling-based unicast/multicast 360° video streaming have been studied [21, 22]. Dynamic navigation-aware multiple 360° video representation based streaming in server-client and edge-based systems have been studied [23–25]. Finally, a related technology, multi-view video streaming enables a viewer to dynamically navigate a remote scene from a discrete set of viewpoints [26–32].

Only two studies to date have examined the integration of mmWave and edge computing for mobile VR [33, 34], though with limited contributions. [33] studies a user clustering strategy to maximize the user field-of-view frame request admission. [34] studies proactive computing and caching of synthesized interactive VR video frames, to minimize the traffic volume of VR gaming. These approaches are heuristic and do not necessarily enable higher VR immersion fidelity for the user. Similarly, only low-quality/low-resolution (4K) 360° content has been considered. These shortcomings considerably penalize the delivered quality of experience.

II. System Models

A. General aspects

In our VR arcade system, illustrated in Figure 1, users receive the 360° video content through wireless VR headsets. The mmWave APs and Wi-Fi router are interlinked with a collocated edge server, where the compressed scalable content resides. The latter is streamed to the users using dual-band Wi-Fi and mmWave transmission.

B. Scalable 360° Dual-Band Video Transmission

We introduce a scalable 360° representation method that synergistically integrates with mmWave and Wi-Fi communication for efficient resource utilization. In particular, we denote the set of tiles at the same spatial location (i, j) in a GOP as a GOP-tile and construct embedded layers of increasing immersion fidelity for each GOP-tile, as illustrated in Figure 2. We carry this out using the scalable extension of the latest video compression standard, denoted as SHVC [35]. GOP (Group Of Pictures) denotes a block of consecutive 360° video frames compressed together, with no reference to other frames.

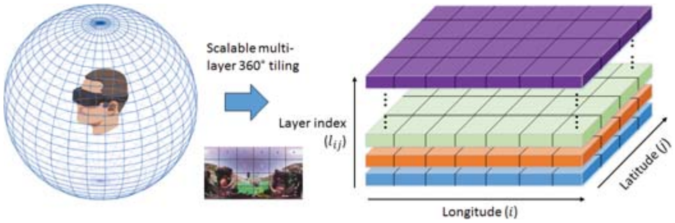

Fig. 2: Scalable 360° tiling. Equirectangular projection is applied first.

The first layer of a compressed GOP-tile is known as the base layer, and the remaining layers are denoted as enhancement layers. The reconstruction fidelity of a GOP-tile improves incrementally as more layers are being decoded progressively starting from the base layer. Let L denote the collection of all GOP-tiles. To augment the system's reliability, and compensate for prospective mmWave link uncertainty and viewport prediction error, the base layer of every GOP-tile $l \in L$ is broadcast over Wi-Fi to all users. Multiple enhancement layers of viewport-specific GOP-tiles are then sent to the individual users over mmWave links. All delivered scalable 360° layers are integrated at a VR user to augment his immersion fidelity.

1) Wi-Fi 360° Base Layer Transmission: Each GOP-tile $l \in L$ exhibits an immersion reconstruction distortion D_l related to its encoding rate R_l as $D_l = a_l R_l^{b_l}$, where a_l and b_l are constants [21]. Let the immersion distortion of each GOP-tile $l \in L$ which is sent over Wi-Fi communication be D_μ and the corresponding data rate of the GOP-tile be R_μ. The bandwidth available for Wi-Fi communication is B_w. Therefore the delay in transmitting all the GOP-tiles $l \in L$ using broadcast is $\tau_c = \frac{\mathcal{L} R_\mu}{B_w \log(1+\gamma_{br})}$, where γ_{br} is the broadcast SNR using Wi-Fi, and $\mathcal{L} = |L|$.

2) mmWave 360° Enhancement Layers Transmission: We leverage our ongoing work on statistical characterization of user viewport navigation [21, 36], to identify L_u as the subset of 360° GOP-tiles that overlap with the VR viewport of user u over that GOP. Essentially, L_u comprises the GOP-tiles that exhibit a non-zero likelihood of being navigated by the user during that GOP of the 360° content. Enhancement layers of tiles $l \in L_u$ will be sent to user u using mmWave transmission from one select AP. In particular, from the L_u set of GOP-tiles to be sent via mmWave AP a, a $L_{u,r} \subseteq L_u$ subset of GOP-tiles are sent raw, while the remaining $L_u \setminus L_{u,r}$ tiles are sent compressed. Let the size of the enhancement layer of each GOP-tile $l \in L_{u,r}$ after decoding be b_r in number of bits and the data rate of each encoded GOP-tile $l \in L_u \setminus L_{u,r}$ be R_l.

In parallel, the user receives the base layer of each GOP-tile $l \in L_u$ (of data rate R_μ) over the Wi-Fi broadcast, as explained earlier. Since we construct/encode the 360° content in scalable manner, the number of bits required to be sent for the enhancement layers of each compressed GOP-tile $l \in L_u \setminus L_{u,r}$ is $(R_l - R_\mu)$. Thus, the delay in transmitting all GOP-tiles $l \in L_u$ using mmWave transmission is $\tau_{a,u} = \frac{|L_{u,r}| b_r + \sum_{l \in L_u \setminus L_{u,r}} (R_l - R_\mu)}{B \log(1+\gamma_{au})}$. The transmitted GOP-tiles $L_u \setminus L_{u,r}$ are decoded at the user, upon reception.

C. Computing time and end-to-end delay analysis

We study the time delay induced by decoding GOP-tiles at the edge server or a user. The time required to decode a GOP-tile depends upon the data rate of the GOP-tile, which in turn depends on the number of scalable layers from which the tile is decoded/reconstructed. When carried out at the edge server, GOP-tile $l \in L_{u,r}$ is decoded from the highest available data rate (best quality) $R_{l,max}$ of the GOP-tile. This corresponds to reconstructing the tile from all K scalable layers into which it has been encoded using our approach. At the user, compressed GOP-tiles $l \in L_{u,r} \setminus L_u$ of data rate R_l (received via mmWave transmission) and GOP-tiles $l \in L$ of data rate R_μ (received via Wi-Fi broadcast) are decoded.

We analyzed the number of CPU computing cycles β required to decode a GOP-tile as a function of its data rate R. We empirically validated a closely-fitting polynomial relationship that we capture as $\beta = cR^3 - dR^2 + eR + f$, where c, d, e, and f are positive constants.

Let the processing capability of the VR headset of user u be f_u, and let $f_{u,1}$ and $f_{u,2}$ be the processing power allocated by the user to decode the GOP-tiles received over Wi-Fi and mmWave, respectively, where $f_{u,1} + f_{u,2} \leq f_u$. Thus, the number of CPU computing cycles required to decode the baseline GOP-tiles L of data rate R_μ at the user is $\mathcal{L} \cdot (cR_\mu^3 - dR_\mu^2 + eR_\mu + f)$. Hence, the induced decoding delay can be formulated as: $T_c = \frac{\mathcal{L} \cdot (cR_\mu^3 - dR_\mu^2 + eR_\mu + f)}{f_{u,1}}$. Next, the aggregate decoding delay for the enhancement GOP-tiles $L_{u,r}$ at the edge server can be formulated as: $T_{u,1} = \frac{\sum_{l \in L_{u,r}} \beta_{l,k_r}}{F_u}$, where $\beta_{l,k_r} = cR_{l,max}^3 - dR_{l,max}^2 + eR_{l,max} + f$ is the number of CPU cycles required to decode tile $l \in L_{u,r}$, and F_u is the edge server's computing resource allocated to user u. Finally, the decoding delay of enhancement layer GOP-tiles $L_u \setminus L_{u,r}$ at the user can be formulated as: $T_{u,2} = \frac{\sum_{l \in L_u \setminus L_{u,r}} cR_l^3 - dR_l^2 + eR_l + f}{f_{u,2}}$. Thus, the aggregate delay, comprising transmission and decoding components, in receiving the GOP-tiles L via Wi-Fi transmission is $T_c + \tau_c$. Similarly, the aggregate delay in receiving the GOP-tiles L_u via a mmWave link is $T_{u,1} + T_{u,2} + \tau_{a,u}$.

III. Problem Formulation

Let Π denote the set of all possible AP to user assignments, for the AP set A and user set U, such that every member set $\pi \in \Pi$ features AP to user assignments comprising $|U|$ disjoint AP user pairs. Furthermore, let $\boldsymbol{L}_u$ be the power set of the set L_u which is the set of all subsets of L_u, including the empty set and L_u itself. Using the advances in [21, 22], we can characterize the likelihood p_l^u of every GOP-tile $l \in L_u$ appearing in the viewport of user u over that GOP, and the expected immersion distortion experienced by the user, given a data rate allocation across the GOP-tiles. Concretely, we formulate the latter quantity as $\sum_{l \in L_{u,r}} p_l^u a_l R_{l,max}^{b_l} + \sum_{l \in L_u \setminus L_{u,r}} p_l^u a_l R_l^{b_l}$. p_l^u capture the expected fraction of the spatial area of a tile being overlapped with the user' viewport during a GOP, and integrate the aspect that equatorial tiles are more likely navigated than polar tiles. Hence, our distortion formulation above corresponds to a tile-level WS-PSNR (Weighted Spherical PSNR) [37]. We aim to minimize the maximum expected immersion distortion over all users, for given system and application constraints. We formulate this problem as:

$$
\begin{aligned}
&\min_{\substack{\pi \in \Pi, \boldsymbol{F}, L_{u,r} \in \boldsymbol{L}_u, \forall u \\ \boldsymbol{R}, f_{u,1}, f_{u,2}}} \max_{u \in U} \sum_{l \in L_{u,r}} p_l^u a_l R_{l,max}^{b_l} + \sum_{l \in L_u \setminus L_{u,r}} p_l^u a_l R_l^{b_l}, \\
&\text{s.t. } \frac{\mathcal{L}(cR_\mu^3 - dR_\mu^2 + eR_\mu + f)}{f_{u,1}} + \frac{\mathcal{L}R_\mu}{r'} \leq \tau,\ u \in U, \\
&\frac{\sum_{l \in L_{u,r}} \beta_{l,k_r}}{F_u} + \frac{\sum_{l \in L_u \setminus L_{u,r}} cR_l^3 - dR_l^2 + eR_l + f}{f_{u,2}} \\
&+ \frac{|L_{u,r}| b_r + \sum_{l \in L_u \setminus L_{u,r}} (R_l - R_\mu)}{r_{a,u}} \leq \tau,\ u \in U, \\
&\sum_{u \in U} F_u \leq F, \quad f_{u,1} + f_{u,2} \leq f_u,\ u \in U,
\end{aligned} \tag{1}
$$

Here, $r' = B_w \log(1 + \gamma_{br})$, $r_{a,u} = B \log(1 + \gamma_{au})$, and B_w and γ_{br} are the bandwidth and SNR of the Wi-Fi broadcast channel. B and γ_{au} denote the respective channel quantities for the mmWave link from AP a to user u. $\boldsymbol{F}$ is the vector of all values of F_u, $u \in U$, $\boldsymbol{R}$ is a set that contains all possible R_l, $l \in L_u \setminus L_{u,r}$, $u \in U$, and τ is the maximum tolerable delay within which each GOP needs to be received such that users do not experience any lag. The first constraint in (1) imposes that the total delay of receiving all baseline GOP-tiles $l \in L$ at the user, via Wi-Fi communication, be bounded by τ. Similarly, the second constraint imposes that the total delay of receiving all enhancement layer GOP-tiles $l \in L$ at the user via mmWave communication be bounded by τ. The computing allocation at the edge server is restricted by the total available resource F, as shown in the third constraint. The restriction on computing resource allocation for each user $u \in U$ is given by the fourth constraint.

Note that minimizing the immersion distortion in (1) is the same as maximizing the respective immersion fidelity, due to the one-to-one mapping between them. Moreover, the (stronger) latency constraints in (1) imply the respective transmission rate constraints on the Wi-Fi and mmWave links over the same time period τ. Other system and application aspects can be easily integrated into (1), e.g., further data processing delays introduced at the edge server and mobile VR users.

IV. Optimization solution

A. Solution outline

The optimization (1) is mixed-integer programming, which is hard to solve optimally in practice, due to its complexity. Thus, we investigate a lower-complexity solution that comprises three steps applied sequentially. First, we fix the edge server's computing resource allocation, to be able to tackle the integer variables in the optimization problem. Then, (1) decomposes into a joint mmWave data rate allocation, user computing resource allocation, and raw tile selection, for each user u to AP a pairing in a given assignment. We compute solutions to each of these independent problems, for any prospective user to AP assignment π, in Section IV-B. These solutions will produce edge weights for any user u to AP a prospective pairing, which we will then leverage in Section IV-C to compute the optimal user to AP assignment π^* using a graph-theoretic solution. Finally, given this assignment and considering now $\boldsymbol{F}$ as a variable, we resolve (1) to jointly identify/update the optimal computing resource allocation at the edge server and users, and the mmWave data rate allocation. This step completes our optimization strategy and is carried out in Section IV-D. Due to limited space, we can only describe the first step in detail here.

B. Computing optimal edge weights for AP to user pairings

Let the edge server's computing resource allocation be fixed and uniform across all users, i.e., $F_u = F/M$, where $M = |U|$ denotes the number of users. Due to the fixed allocation, (1) decouples into:

$$
\begin{aligned}
&\min_{\substack{L_{u,r} \in \boldsymbol{L}_u \boldsymbol{R} \\ f_{u,1}, f_{u,2}}} \sum_{l \in L_{u,r}} p_l^u a_l R_{l,max}^{b_l} + \sum_{l \in L_u \setminus L_{u,r}} p_l^u a_l R_l^{b_l}, \\
&\text{s.t. } \frac{\mathcal{L}(cR_\mu^3 - dR_\mu^2 + eR_\mu + f)}{f_{u,1}} + \frac{\mathcal{L}R_\mu}{r'} \leq \tau, \\
&\frac{\sum_{l \in L_{u,r}} \beta_{l,k_r}}{F_u} + \frac{\sum_{l \in L_u \setminus L_{u,r}} cR_l^3 - dR_l^2 + eR_l + f}{f_{u,2}} \\
&+ \frac{|L_{u,r}| b_r + \sum_{l \in L_u \setminus L_{u,r}} (R_l - R_\mu)}{r_{a,u}} \leq \tau,\ f_{u,1} + f_{u,2} \leq f_u,
\end{aligned} \tag{2}
$$

for each (user u, AP a) pairing in an assignment π. To solve (2), we first fix the set of raw enhancement layer GOP-tiles $L_{u,r}$, and formulate an optimization strategy to solve the allocation of mmWave data rate and user computing resource. Then, we show how to integrate the selection of $L_{u,r}$ into our optimization strategy, by reformulating (2) accordingly.

1) Fixed set of raw GOP-tiles: We first solve (2), for a given $L_{u,r}$. We can show that this problem is not convex. It can be reformulated into geometric programming (GP) via the single condensation method [38]. According to this method, for a constraint which is a ratio of posynomials, the denominator posynomial (say $\mathrm{f}(\mathbf{x})$) can be approximated into a monomial using the following inequality: $\mathrm{f}(\mathbf{x}) = \sum_\ell \mathrm{f}_\ell(\mathbf{x}) \geq \hat{\mathrm{f}}(\mathbf{x}) = \prod_\ell \left[\frac{\mathrm{f}_\ell(\mathbf{x})}{\delta_\ell}\right]^{\delta_\ell}$ (3), where $\delta_\ell > 0$ and $\sum_\ell \delta_\ell = 1$. Then, for $\delta_\ell = \mathrm{f}_\ell(\hat{\mathbf{x}})/\mathrm{f}(\hat{\mathbf{x}})$, $\hat{\mathrm{f}}(\hat{\mathbf{x}})$ is the best monomial approximation of $\mathrm{f}(\mathbf{x})$ near $\mathbf{x} = \hat{\mathbf{x}}$.

We formulate an iterative technique to optimally solve (2) in this case. In particular, at each iteration t, the first constraint in (2) is converted into a posynomial using (3). Similarly, at every iteration t, we can convert the second constraint in (2) into a posynomial using (3). Space limits prevent including these expressions here. Let $D(t) = \sum_{l \in L_{u,r}} p_l^u a_l R_{l,max}^{b_l} + \sum_{l \in L_u \setminus L_{u,r}} p_l^u a_l R_l(t)^{b_l}$. Then, the overall optimization to solve at iteration t is

$$
\min_{\boldsymbol{R}, f_{u,1}, f_{u,2}} D(t), \text{ s.t. convert. constr., } f_{u,1}(t) + f_{u,2}(t) \leq f_u. \tag{4}
$$

The above optimization problem is GP and can be solved optimally. The iterative optimization is carried out until $|D(t) - D(t-1)| \leq \epsilon$, with $0 \leq \epsilon \ll 1$. An algorithmic implementation is included in Algorithm 1, which converges to the global solution [38].

Algorithm 1 GP based solution for (2), for a given $L_{u,r}$.

```
Set $t = 1$,
Initialize $f_{u,1}(t) = f_{u,2}(t) = f_u/2$, Initialize $R_l(t)$
while true do                                    ▷ infinite loop
    $t = t + 1$
    Calculate $\delta_1(t)$, $\delta_2(t)$
    Find the optimum $f_{u,1}(t)$, $f_{u,2}(t)$ $D(t)$, $R_l(t)$ solving (4) using GGPLAB [39]
    if $|D(t) - D(t-1)| \leq \epsilon$ then
        Break
    end if
end while
The optimal value of the optimization problem (2) is $D(t)$
```

2) Raw GOP-tile selection: The optimization problem in (2) can be solved optimally in the following manner: (i) For each possible tile set $L_{u,r} \in L$, find the mmWave data rate and user computing resource allocation by solving (2), and then (ii) Find the best tile set $L_{u,r}$ for which the expected immersion distortion is smallest. However, this scheme requires searching over $2^{|L|} - 1$ possible tile sets. To solve (2) with low complexity, first, we reformulate our problem as follows: Let x_l be an indicator function that denotes whether a tile is sent as encoded or raw, where $x_l = 1$, if tile l is sent as encoded, and $x_l = 0$, if the tile is sent as raw. We then reformulate (2) accordingly, omitting the resulting expression here to save space. To solve the reformulated problem efficiently, we first replace the binary constraints above with continuous equivalents $x_l \in [0,1], l \in L$. We can then convert the resulting problem into GP and solve it.

The obtained optimal solution for x_l is continuous. To find the desired raw GOP-tile selection, we pursue the following rounding strategy. We first initialize $L_{u,r}$ as empty. Then, at each step: (i) We find the tile l^* with smallest value of x_l among the available tile set $L \setminus L_{u,r}$, and (ii) If the expected immersion distortion reduces, add l^* to the raw tile set $L_{u,r}$. We continue this process as long as the immersion distortion reduces further, and finally we produce the desired tile set $L^*_{u,r}$, at the end. Then, the allocation of mmWave data rate and user computing resource can be obtained by solving (2), for the given $L^*_{u,r}$. Let $D^*_{a,u}$ denote the optimal expected immersion distortion experienced by user u, i.e., the value of the objective function in (2), enabled by the produced solution. Algorithm 2 summarizes formally our optimization procedure.

C. Optimal user to access point assignment

The optimal user to AP assignment can be identified by searching over all possible assignments Π. But, this requires searching over $(M+N)!/M!$ prospective assignments, where $N = |A|$ is the number of APs. We explore a lower-complexity alternative that uses graph theoretic concepts. To solve the AP to user assignment, we first formulate a weighted bipartite graph in which each AP $a \in \{1,..,N\}$ and each user $u \in \{1,..,M\}$ are represented by vertices $v^1_a \in \mathcal{V}^1$ and $v^2_u \in \mathcal{V}^2$, respectively, and the weight of the edges (v^1_a, v^2_u) is expressed as $\omega_{(v^1_a, v^2_u)} = D^*_{a,u}$.

Leveraging the development heretofore, we formulate and solve the user to AP assignment subproblem from (1) as a bottleneck matching problem for the graph, defined by the maximum matching whose largest edge weight is as small as possible, i.e., $\min_{\phi \in \Phi} \max_{(v^1_a, v^2_u) \in \phi} \omega_{(v^1_a, v^2_u)}$, where Φ contains all possible maximum matchings. Φ is directly related to Π such that each maximum matching $\phi \in \Phi$ corresponds to a user to AP assignment in Π.

Algorithm 2 Optimal solution of (2), with selection of $L^*_{u,r}$.

```
Set $t = 1$, $L_{u,r} = \emptyset$
Solve reformulated (2)
for $i = 1 : |L|$ do
    $l^* = \arg\min_{l \in L \setminus L_{u,r}} x_l$
    $L_{u,r} = L_{u,r} \cup l^*$
    Solve (2) considering raw tile set $L_{u,r} \cup l^*$ and obtain $D_{a,u}(L_{u,r} \cup l^*)$
    if $D_{a,u}(L_{u,r} \cup l^*) <= D_{a,u}(L_{u,r})$ then
        $L_{u,r} = L_{u,r} \cup l^*$
    else
        break
    end if
end for
Output raw tile set $L^*_{u,r}$
Solve (2), given $L^*_{u,r}$. Compute objective $D^*_{a,u}$.
```

D. Joint Transmission and User/Edge Resource Allocation

In Section IV-B, as part of the optimization carried out therein, we identified the optimal enhancement GOP-tile subset $L^*_{u,r}$ that should be transmitted as raw data over the mmWave link of a given user and AP pairing (u,a). In Section IV-C, we identified the optimal user to AP assignment π^*. Given these discrete optimization developments, we can (re)solve jointly now the optimal allocation of user and edge server computing resources, and mmWave link data rate across the compressed enhancement layer GOP-tile subset $L_u \backslash L^*_{u,r}$. Concretely, we investigate the joint allocation of these three system resources by solving (1), for given π^* and $L^*_{u,r}, \forall u$.

We pursue a solution to this optimization problem by reformulating it first as GP using the single condensation method, analogously to Section IV-B1. We then solve the problem reformulation via an iterative optimization method equivalent to Algorithm 1.

V. Experimental results

We carry out an experimental evaluation of the performance of our system. We measure the delivered VR immersion fidelity as the inverse of the respective distortion quantity, commonly known as the Peak Signal-to-Noise ratio (PSNR). We benchmark our approach relative to a state-of-the-art method that integrates the latest video streaming standard MPEG DASH [18, 40], to deliver the 360° content over Wi-Fi, given the same system constraints. No prior work has considered our problem setting, to serve as another reference method. In our simulation experiments, five users are uniformly distributed in a $5m \times 5m$ square room VR arcade. The mmWave APs and the Wi-Fi router, linked to an edge computing server, are placed high on the room walls. As 360° content, we use the 'Runner' and 'Basketball' 360° videos captured at 30fps and 8K spatial resolution [41].

A. Immersion Fidelity vs. Edge Computing Resource

In Figure 3, we examine the performance of the proposed system when the edge server computing power varies from 75 GHz to 175 GHz. The mmWave network links in the system exhibit diverse data rates in the range $600-900$ Mbps. We observed in this experiment that as the server's computing power increases, more enhancement layer GOP-tiles can be decoded within a small computing latency at the edge, which in turn reduces the computing latency at the user. Thus, the mmWave APs can send a higher number of raw enhancement GOP-tiles, and compressed enhancement GOP-tiles, encoded at higher data rates, which improves the PSNR. We observe consistently strong performance improvement of 8-9 dB and 9-10 dB over MPEG-DASH for Runner and Basketball, respectively. These advances will considerably enhance the immersion fidelity and quality of experience delivered to mobile users in our VR system.

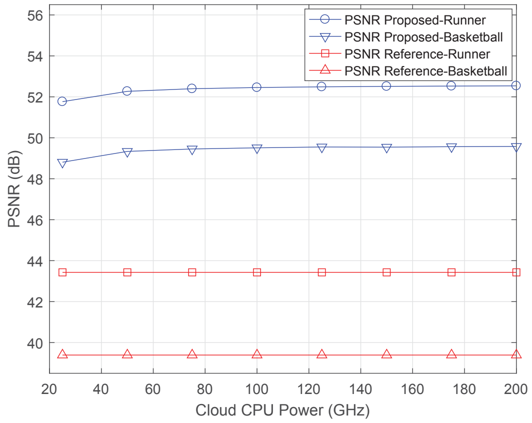

Fig. 3: PSNR performance versus edge server computing power.

B. *Immersion Fidelity vs. mmWave Transmission Rate*

Next, in Figure 4 we explore the performance of our system, when the data rate of the mmWave network links in the system is uniform, and is progressively increased from 400 Mbps to 1 Gbps. The edge server's computing power is fixed to 150 GHz in these experiments. We can see that as the mmWave transmission data rate increases, the PSNR of the proposed system increases, as expected. In particular, as the data rate increases, more raw enhancement layer GOP-tiles can be transmitted. In consequence, this will reduce the computing time at the user, which together with the higher mmWave network link data rate, will enable transmitting the remaining compressed enhancement GOP-tiles, encoded at higher data rates. Again, we observe consistently strong performance improvement of 7-9 dB and 8-10 dB over MPEG-DASH for Runner and Basketball, respectively. These benefits considerably advance the state-of-the-art.

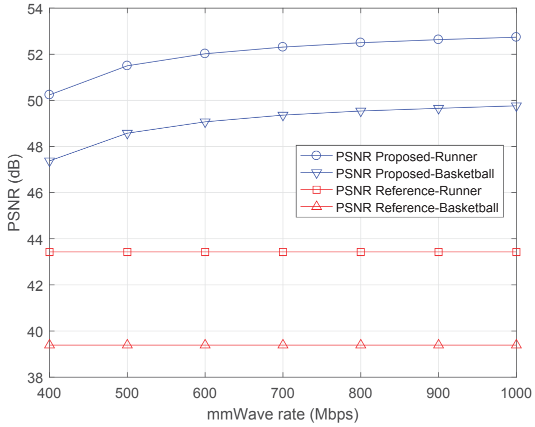

Fig. 4: PSNR versus mmWave rate ($F = 150$ GHz).

C. *Expected number of selected raw enhancement layer GOP-tiles*

Finally, in Table I and Table II, we investigate the expected number of raw enhancement layer GOP-tiles selected for mmWave transmission in our system, in the case of the 360° content 'Runner'. A positive non-integer valued entry in Table I, associated with a given mmWave transmission data rate and edge server computing power pair, can be explained with the following examples.

TABLE I: Expected number of enhancement layer raw GOP-tiles selected for mmWave transmission vs. edge server computing power and mmWave transmission data rate. User computing power $f_u = 3$ GHz.

mmWave rate	400 Mbps	600 Mbps	800 Mbps	1000 Mbps
$F = 50$ GHz	0	0	0.6	1.1
$F = 100$ GHz	0	0	0.8	1.3
$F = 150$ GHz	0	0.2	1.0	1.6
$F = 200$ GHz	0	0.4	1.2	1.9

(i) if $F = 150$ GHz and mmWave rate is 600 Mbps, an expected number of transmitted raw tiles of 0.2 can occur if for one of the five users, one enhancement layer GOP-tile is transmitted as raw data. For the other users, all enhancement GOP-tiles are transmitted compressed. (ii) if $F = 200$ GHz and mmWave rate is 800 Mbps, an expected number of transmitted raw tiles of 1.2 can occur if for one of the five users, two enhancement GOP-tiles are transmitted as raw data. For the other four users, one enhancement GOP-tile is transmitted as raw data.

It can be observed from Table I that as the mmWave transmission data rate and edge server computing power increase, a higher number of raw enhancement GOP-tiles are selected by our optimization framework, to augment the delivered immersion fidelity. This is because with the increase in mmWave data rate and edge server computing power, a higher number of enhancement GOP-tiles can be decompressed at the edge server and transmitted as raw data over the mmWave link, within a short time interval. In consequence, then only a smaller number of compressed enhancement GOP-tiles, encoded at higher data rate, would need to be delivered, which would lower the decoding latency induced at the user. Both of these advances will contribute to higher immersion fidelity delivered to the VR user, while maintaining the required system latency.

TABLE II: Expected number of enhancement layer raw GOP-tiles selected for mmWave transmission vs. user computing power and mmWave transmission data rate. Edge server computing power $F = 150$ GHz.

mmWave rate	400 Mbps	600 Mbps	800 Mbps	1000 Mbps
$f_u = 3$ GHz	0	0.2	1.0	1.6
$f_u = 4$ GHz	0	0.2	1.0	1.7
$f_u = 5$ GHz	0	0.2	1.1	1.8
$f_u = 6$ GHz	0	0.3	1.2	1.8

We can observe from Table II that a higher number of raw enhancement GOP-tiles are likewise selected for transmission, as the mmWave transmission data rate and user computing power increase, again to augment the immersion fidelity delivered to the user. This outcome stems from reasons equivalent to those discussed earlier in the context of the results presented in Table I. In particular, the higher user computing power enables decoding faster compressed enhancement GOP-tiles delivered to the user, i.e., with lower induced delay. This in turn will leave more of the end-to-end system delay constraint available to be consumed by mmWave transmission, which coupled with the higher mmWave transmission data rate, enables sending a higher number of raw enhancement GOP-tiles.

VI. Conclusion

We presented a novel mobile VR system for streaming scalable 8K 360° video at high reliability and immersion fidelity, and low interactive latency, via the synergistic integration of embedded 360° tiling, dual millimeter wave (mmWave) and Wi-Fi transmission, and edge computing. We explored the joint optimization of the mmWave access point to user association, the choice of enhancement layer 360° tiles to be transmitted decompressed, the allocation of mmWave data rate across the compressed tiles in a viewport-specific enhancement layer, and the allocation of computing resources at the edge server and user devices. Our objective was to maximize the minimum delivered VR immersion fidelity across all users, given transmission, latency, and computing constraints. We formulated a solution that comprises multiple geometric programming algorithms and an intermediate step of graph-theoretic VR user to mmWave AP assignment. We demonstrated considerable gains in delivered immersion fidelity (8 dB to 10 dB) and spatial resolution (8K vs. 4K), over MPEG-DASH that uses Wi-Fi transmission only. We have shown that an increasing number of raw 360° enhancement layer GOP-tiles are sent, as the mmWave link data rate or the edge server's computing power increases, exploring rigorously here the fundamental trade-offs between computing and communication capabilities, end-to-end system latency, and delivered immersion fidelity.

References

[1] Digi-Capital, "Ubiquious 90 billion AR to dominate focused 15 billion VR by 2022", Jan. 2018.

[2] J. Chakareski and P. Frossard, "Distributed collaboration for enhanced sender-driven video streaming," *IEEE Trans. Multimedia*, Aug. 2008.

[3] N. Thomos, J. Chakareski, and P. Frossard, "Randomized network coding for UEP video delivery in overlay networks," in *Proc. IEEE Int'l Conf. Multimedia and Exhibition (ICME)*, July 2009.

[4] J. Chakareski, J. Apostolopoulos, S. Wee, W.-T. Tan, and B. Girod, "R-D hint tracks for low-complexity R-D optimized video streaming," in *Proc. IEEE Int'l Conf. Multimedia and Exhibition*, Jun. 2004.

[5] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. IEEE Data Compression Conference*, Mar. 2003.

[6] A. B. Reis, J. Chakareski, A. Kassler, and S. Sargento, "Distortion optimized multi-service scheduling for next generation wireless mesh networks," in *Proc. IEEE INFOCOM Int'l Workshop on Carrier-grade Wireless Mesh Networks*, Mar. 2010.

[7] J. Chakareski, J. Apostolopoulos, W.-T. Tan, S. Wee, and B. Girod, "Distortion chains for predicting the video distortion for general packet loss patterns," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 5. Montreal, Canada: IEEE, May 2004, pp. 1001–1004.

[8] J. Chakareski and P. Frossard, "Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources," *IEEE Trans. Multimedia*, vol. 8, no. 2, Apr. 2006.

[9] J. Chakareski and B. Girod, "Rate-distortion optimized video streaming with rich acknowledgments," in *Proc. SPIE Conf. on Visual Communications and Image Processing*, San Jose, CA, Jan. 2004, pp. 661–668.

[10] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," *ACM/Springer-Verlag Multimedia Systems Journal*, no. 1, Jan. 2005.

[11] B. Begole, "Why the Internet pipes will burst when virtual reality takes off," *Forbes Magazine*, Feb. 2016.

[12] J. Chakareski, "Drone networks for virtual human teleportation," in *Proc. ACM MobiSys Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DroNet)*, Jun. 2017.

[13] ——, "UAV-IoT for next generation virtual reality," *IEEE Trans. Image Processing*, vol. 28, no. 12, pp. 5977–5990, Dec. 2019.

[14] ——, "Aerial UAV-IoT sensing for ubiquitous immersive communication and virtual human teleportation," in *Proc. INFOCOM Workshop Communication & Networking Techniques for Contemporary Video*, May 2017.

[15] E. Cuervo, K. Chintalapudi, and M. Kotaru, "Creating the perfect illusion: What will it take to create life-like virtual reality headsets?" in *Proc. ACM Int'l Workshop Mobile Computing Systems and Applications (HotMobile)*, Tempe, AZ, USA, Feb. 2018, pp. 7–12.

[16] M. Champel, T. Stockhammer, T. Fautier, E. Thomas, and R. Koenen, "Quality requirements for VR," in *Proc. 116th MPEG Meeting of ISO/IEC JTC1/SC29/WG11*, Oct 2016.

[17] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *Proc. IEEE ISM*, Dec 2016, pp. 107–110.

[18] M. Graf, C. Timmerer, and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over HTTP: Design, implementation, and evaluation," in *Proc. ACM Multimedia Systems Conference*, Taipei, Taiwan, June 2017, pp. 261–271.

[19] A. Zare, A. Aminlou, M. Hannuksela, and M. Gabbouj, "HEVC-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proc. ACM Multimedia Conference*, Oct. 2016.

[20] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge, and T. Schierl, "HEVC tile based streaming to head mounted displays," in *Proc. IEEE Consumer Communications & Networking Conference*, Jan 2017, pp. 613–615.

[21] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, "Viewport-driven rate-distortion optimized 360° video streaming," in *Proc. IEEE Int'l Conf. Communications*, May 2018, pp. 1–7.

[22] R. Aksu, J. Chakareski, and V. Swaminathan, "Viewport-driven rate-distortion optimized scalable live 360° video network multicast," in *Proc. IEEE ICME Hot 3D Workshop*, July 2018, pp. 1–6.

[23] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. IEEE Int'l Conf. Communications*, Paris, France, May 2017, pp. 1–7.

[24] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski, "Optimal set of 360-degree videos for viewport-adaptive streaming," in *Proc. ACM Multimedia Conference*, Mountain View, CA, USA, October 2017.

[25] J. Liu, G. Simon, X. Corbillon, J. Chakareski, and Q. Yang, "Delivering viewport-adaptive 360-degree videos in cache-aided MEC networks," in *Proc. IEEE Int'l Symp. on Broadband Multimedia Systems and Broadcasting*, Paris, France, Jun. 2020.

[26] J. Chakareski, V. Velisavljević, and V. Stanković, "User-action-driven view and rate scalable multiview video coding," *IEEE Trans. Image Processing*, vol. 22, no. 9, pp. 3473–3484, Sep. 2013.

[27] ——, "View-popularity-driven joint source and channel coding of view and rate scalable multi-view video," *IEEE J. Selected Topics in Signal Processing*, vol. 9, no. 3, pp. 474–486, Apr. 2015.

[28] J. Chakareski, "Wireless streaming of interactive multi-view video via network compression and path diversity," *IEEE Trans. Communications*, vol. 62, no. 4, pp. 1350–1357, Apr. 2014.

[29] ——, "Uplink scheduling of visual sensors: When view popularity matters," *IEEE Trans. Communications*, vol. 2, no. 63, Feb. 2015.

[30] ——, "Transmission policy selection for multi-view content delivery over bandwidth constrained channels," *IEEE Trans. Image Processing*, vol. 23, no. 2, pp. 931–942, Feb. 2014.

[31] L. Rossi, J. Chakareski, P. Frossard, and S. Colonnese, "A Poisson Hidden Markov Model for multiview video traffic," *IEEE/ACM Trans. Networking*, vol. 23, no. 2, pp. 547–558, Apr. 2015.

[32] Z. Liu, G. Cheung, J. Chakareski, and Y. Ji, "Multiple description coding and recovery of free viewpoint video for multi-path wireless network streaming," *IEEE J. Selected Topics in Signal Processing*, Feb. 2015.

[33] C. Perfecto, M. Elbamby, J. D. Ser, and M. Bennis, "Taming the latency in multi-user VR 360°: A QoE-aware deep learning-aided multicast framework," *Arxiv*, Nov. 2018.

[34] 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 *Proc. IEEE WCNC*, April 2018, pp. 1–6.

[35] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of SHVC: Scalable extensions of the high efficiency video coding standard," *IEEE Trans. Circuits Systems for Video Technology*, Jan. 2016.

[36] X. Corbillon, F. D. Simone, and G. Simon, "360-degree video head movement dataset," in *Proc. ACM Multimedia Systems Conf.*, Jun. 2017.

[37] Y. Sun, A. Lu, and L. Yu, "Weighted-to-spherically-uniform quality evaluation for omnidirectional video," *IEEE Signal Processing Letters*, vol. 24, no. 9, pp. 1408–1412, Sep. 2017.

[38] G. Xu, "Global optimization of signomial geometric programming problems," *Eur. J. Oper. Res.*, vol. 233, no. 3, pp. 500–510, 2014.

[39] GGPLAB: A simple MATLAB toolbox for geometric programming. [Online]. Available: http://www.stanford.edu/boyd/ggplab/

[40] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[41] X. Liu, Y. Huang, L. Song, R. Xie, and X. Yang, "The SJTU UHD 360-degree immersive video sequence dataset," in *Proc. IEEE Int'l Conf. Virtual Reality and Visualization*, Zhengzhou, China, Oct 2017.