Viewport-Adaptive Scalable Multi-User Virtual Reality Mobile-Edge Streaming

Jacob Chakareski, *Senior Member, IEEE*

***Abstract*— Virtual reality (VR) holds tremendous potential to advance our society, expected to make impact on quality of life, energy conservation, and the economy. To bring us closer to this vision, the present paper investigates a novel communications system that integrates for the first time scalable multi-layer 360° video tiling, viewport-adaptive rate-distortion optimal resource allocation, and VR-centric edge computing and caching, to enable next generation high-quality untethered VR streaming. Our system comprises a collection of 5G small cells that can pool their communication, computing, and storage resources to collectively deliver scalable 360° video content to mobile VR clients at much higher quality. The major contributions of the paper are the rigorous design of multi-layer 360° tiling and related models of statistical user navigation, analysis and optimization of edge-based multi-user VR streaming that integrates viewport adaptation and server cooperation, and base station 360° video packet scheduling. We also explore the possibility of network coded data operation and its implications for the analysis, optimization, and system performance we pursue in this setting. The advances introduced by our framework over the state-of-the-art comprise considerable gains in delivered immersion fidelity, featuring much higher 360° viewport peak signal to noise ratio (PSNR) and VR video frame rates and spatial resolutions.**

***Index Terms*— Mobile virtual reality, scalable 360° video tiling, mobile edge computing and streaming, resource allocation, 5G small cell systems, statistical VR navigation analysis, multiple knapsack problem with multiple constraints, branch-and-prune fully-polynomial time approximation method.**

I. Introduction

VIRTUAL reality holds tremendous potential to advance our society. It is expected to make impact on quality of life, energy conservation, and the economy [1], [2], and reach a $162B market by 2020 [3]. As the Internet-of-Things (IoT) is becoming a reality, modern technologists envision transferring remote contextual and environmental immersion experiences as part of an online VR session. However, two main highly-intertwined challenges stand in the way of realizing this vision: VR requires **(1) ultra-low latency high data rate communications, and (2) highly data-intensive computing**. Neither of these challenges can be met by current and upcoming traditional communications systems [4], [5], as the content to be delivered is too voluminous and the headsets' computing/storage capabilities are insufficient within an acceptable and wearable form factor. Hence, VR applications are presently limited to off-line operation, low-fidelity graphics content, tethered high-end computing equipment, and gaming/entertainment settings. 360° video is the first actual-scene content format to enable remote VR immersion. However, emerging 360° streaming practices are highly inefficient, which considerably degrades the quality of experience, as explained in detail later.

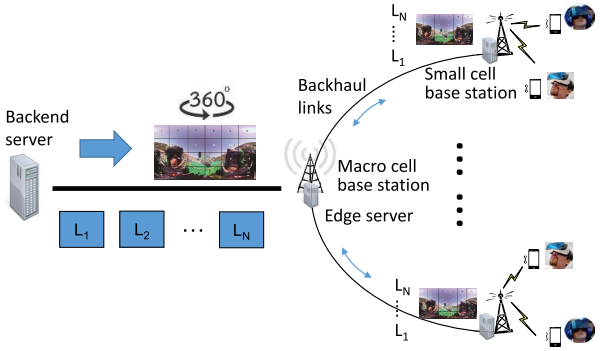

Fig. 1. System scenario under investigation.

To overcome these challenges, we investigate for the first time a novel communications system that integrates scalable multi-layer 360° video tiling, viewport-adaptive rate-distortion optimal resource allocation, and VR-centric edge computing and caching, to enable next generation high-quality untethered on-demand VR streaming. Our system is illustrated in Figure 1 and comprises a collection of 5G small cells featuring a base station and an edge server each, which pool their communication, computing, and storage resources to collectively deliver scalable 360° video content to mobile VR clients, at much higher quality. Cooperation among the small cells is enabled via backhaul links that interconnect them, and the scalable 360° content featuring multiple layers L_i of incrementally increasing quality is initially delivered from a backend server, as illustrated in Figure 1. Considerable advances in 360° video quality, frame rate, and spatial resolution are enabled.

The rest of the paper is organized as follows. We first provide some background on 360° video streaming and the related challenges that arise therein. Overview of related work is provided in Section III. The framework of our system is then presented over the next three sections. The design of multi-layer 360° tiling and formulation of related models of statistical user navigation are carried out in Section IV. The analysis of edge-based multi-user VR streaming that integrates viewport adaptation and server cooperation is carried out in Section V. Here, we also investigate the possibility of network coded data operation and its implications for the analysis, optimization, and system performance we pursue. Analysis of

Manuscript received October 15, 2019; revised April 3, 2020; accepted April 4, 2020. Date of publication May 6, 2020; date of current version May 29, 2020. This work was supported in part by the NSF under Awards CCF-1528030, ECCS-1711592, CNS-1836909, and CNS-1821875, by the NIH under Award R01EY030470, and by research gifts and an Adobe Data Science Award from Adobe Systems. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Francesco G. B. De Natale.

The author is with the Ying Wu College of Computing, New Jersey Institute of Technology, Newark, NJ 07103 USA (e-mail: jakov@jakov.org).

Digital Object Identifier 10.1109/TIP.2020.2986547

effective scheduling of buffered data packets at the small base stations is carried out in Section VI. Finally, an experimental evaluation of our framework is provided in Section VII and the paper concludes in Section VIII.

II. 360° Video VR Streaming Background

360° video is an emerging video format that is captured by an omnidirectional camera that records incoming light rays from every direction (see Figure 2 top left).

It enables a three dimensional 360° look-around of the surrounding scene for a remote user, virtually placed at the camera location, on his VR head-mounted display (HMD), as illustrated in Figure 2 right. After capture, the spherical 360° raw video frames are first mapped to a wide equirectangular panorama (illustrated in Figure 2, bottom left) and then compressed using state-of-the-art (planar) video compression such as HEVC. The former intermediate step is introduced, as compression techniques operating directly on spherical data are much less mature and performing relative to conventional video compression operating on planar (2D) video frames.

For remote service, when the user and the stored 360° data are not collocated, the entire monolithic 360° panorama is streamed to the user presently, leveraging state-of-the-art video streaming (DASH - Dynamic Adaptive Streaming over HTTP [6]). However, at any point of time, the user experiences only a small portion of it denoted as V_c (current viewport). This considerably penalizes the quality of experience, due to the overwhelming volume of 360° data that needs to be delivered, which exceeds the available network streaming bandwidth C by orders of magnitude. Thus, only very low-quality low-resolution 360° videos can be delivered online presently over the Internet. Similarly, the streaming also lacks the ultra-low latency interactivity required for truly immersive experiences, due to the use of traditional server-client Internet architectures.

Moreover, in the wireless setting, the available data rates C are even lower, and the VR headsets' computing and storage capabilities are insufficient within an acceptable and wearable form factor, to enable independent untethered operation. The same holds even when they are attached to a mobile device, as the latter is also limited in terms of storage and computing capabilities, e.g., high-end mobile GPUs lag their desktop counterparts in computing power by a factor of ten and will not have the required TFlops to provide the necessary rendering computation for high-quality VR anytime soon [7].

The above challenges motivate novel communications strategies and systems that synergistically integrate viewport-adaptive 360° streaming, mobile edge computing and caching, and scalable multi-layer 360° tiling, for next generation untethered VR applications. This is the objective we pursue.

III. Related Work

Cooperative edge-based multi-user mobile VR streaming is a *novel topic*. Related areas include multi-camera wireless sensing [8], immersive telecollaboration [9], [10], multi-view video coding [11]–[13], and 360° Internet streaming [14], [15].

Similarly to our approach, a few existing studies of single-user on-demand 360° Internet streaming [16]–[18] considered splitting the 360° video into spatial tiles as part of the encoding, using the tiling feature of the latest High Efficiency Video Coding (HEVC) standard [19]. However, their design choices are heuristic and lack analysis of the fundamental trade-offs between delivered immersion fidelity, user navigation patterns, coding efficiency, view switching capability, and available system resources, as we carry out. Moreover, our integration of scalability, edge-based delivery, and formal 360° partitioning considerably enhances the VR application interactivity, by reducing its streaming latency, relative to these studies.

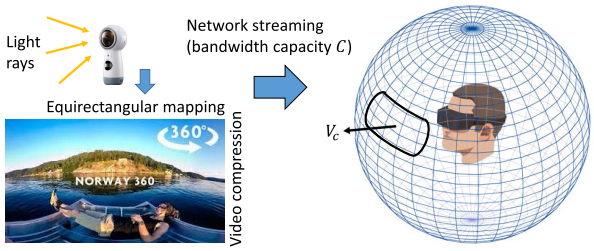

Fig. 2. 360° video capture and streaming, and user viewport V_c.

Existing work on wireless base station caching includes [20], which considered the problem of estimating the content popularity at a base station and minimizing the total delay of content retrieval, formulating the latter as a knapsack problem [21]. Similarly, [22] considered the problem of reducing the delay of content delivery using caching at wireless helper nodes, small-cell base stations that have high storage capability and low coverage, differentiating available helpers based on their proximity to the served node. Moreover, [23] considered optimizing the parameters of a single base station cache and [24], [25] studied hierarchical caching in cellular back-haul networks. Information-theoretic studies of hierarchical caching are carried out in [26], [27]. Joint caching and channel assignment in multi-cellular systems is studied in [28], [29].

This paper has been inspired by a short preliminary study we carried out earlier in [30]. The major technical novelties the paper introduces relative to [30] include:

- Rigorous design/formulation of scalable multi-layer 360° tiling and related statistical models of VR user navigation.
- Viewport-adaptive analysis and optimization of the rate-distortion trade-offs and resource allocation over the 360° video panorama that integrate the above advances, for efficient single-user 360° video streaming.
- Edge-based multi-user VR streaming that integrates server cooperation, VR edge computing/caching, and the above advances, and introduces more general/accurate system models, analysis, and optimization methods. Thereby, the fundamental performance trade-offs between system resources, cost, and delivered VR user immersion fidelity that arise in this context are captured and understood more precisely.
- Rigorous analysis of prospective networking coding data packet operation in the above context and its implications for the resource optimization and system performance.
- Effective small base station 360° video packet scheduling to address prospective network transients and buffering.
- Comprehensive performance evaluation of our system framework and its major components demonstrating considerable gains in delivered VR immersion fidelity, featuring

Fig. 3. Example 6×4 spatial tiling of a 360° panorama.

much higher 360° viewport peak signal to noise ratio (PSNR) and VR video frame rates and spatial resolutions.

IV. Scalable Multi-Layer 360° Tiling

Tiling of a wide-panorama video has been introduced as an option in HEVC [19], to facilitate parallel processing of the tiled segments of the video in multi-core processor systems. In our case, we leverage tiling of the 360° panorama, to capture effectively the user viewport over time and exploit the uneven rate-distortion trade-offs that arise across the spatial panorama. Moreover, tiling the 360° panorama will also facilitate developing a statistical model of user navigation, as explained in Section IV-B. These three aspects will then be integrated effectively into an analysis that dynamically selects the amount of resources allocated over space and time, for a 360° video as it is being streamed to a user. An effective implementation of the analysis is enabled via a scalable multi-layer 360° tiling. The design of the scalable tiling and the respective analysis are formulated in Section IV-D.

A. 360° Panorama Tiling

We partition each video frame of a 360° video into a set of $N \times M$ spatial tiles, as illustrated in Figure 3, where the first and second dimensions of the denoted tiling, (N, M), parallel the horizontal and vertical spatial dimensions of the 360° video frames. Each tile is then independently encoded and streamed to the user, according to our analysis and optimization. In Figure 3, the tiles are indexed/labeled in a raster fashion, top-to-bottom and left-to-right.

Denser tiling layouts increase the processing complexity and reduce the compression efficiency, but, enable more precise delineation of the user viewport and thus more efficient viewport-aware resource allocation across the 360° panorama.

In our work, we have empirically observed that the 6×4 and 8×6 tiling options provide good performance in terms of processing complexity and compression efficiency, as induced by the selected tiling, for the 360° video spatial resolutions available today (4K and 8K). Similar observation has been made recently in [31] in the case of 4K 360° videos. Independently, a VR spin-off reported using around 100 tiles for 8K 360° panoramas without severe compression inefficiency [32]. Selecting the optimal $N \times M$ tiling, where N and M can be arbitrary integer numbers, is an NP hard problem, due to its discrete combinatorial nature. Similarly, selecting the optimal adaptive tiling, where the number and size of tiles across the spatial panorama can be varied, akin to the selection of encoding macro-block and block sizes in modern video encoders, is also an NP hard problem [33]. Brute-force approaches of evaluating a large number of prospective tiling options, in the pursuit of this objective, with the prospective facility of large deep neural networks, have been attempted in [33], however, without clear evidence of convincing performance benefits to justify the required huge investment in computing complexity.

B. 360° Navigation Data Capture

We have captured navigation data that characterizes how a mobile VR user explores a 360° video over time. Specifically, his VR head-mounted display (device) outputs the direction of the current viewpoint of the user V_c on the 360° view sphere up to 250 times per second (see Figure 2 right, for an illustration). Formally, this is the surface normal of V_c on the 360° sphere, which is uniquely characterized by the azimuth and polar angles $\varphi \in [0°, 360°]$ and $\theta \in [0°, 180°]$ that it spans on the sphere, in a spherical coordinate system with the 360° sphere center as its origin (see Figure 4, right). These two angles are equivalently denoted as yaw and pitch in the VR community, captured as rotation angles around the Z and Y axes (see Figure 4, left). We recorded the (φ_j, θ_j) pairs that coincided with the discrete temporal instances t_j of consecutive 360° video frames j from which the respective viewport V_c is selected to be displayed to the user, as he navigates the content. We leverage this data to formulate our statistical analysis of user navigation in the next section.

C. Statistical Characterization of User Navigation

Let the set $\{(\varphi_j, \theta_j)\}$ denote a navigation trace for a given 360° video and VR user. Let $S_j^{V_c}$ denote the set of pixels in the 360° panorama occupied by the user viewport V_c at time instance t_j (temporal video frame j). Similarly, let S_j^{nm} denote the set of pixels in the 360° panorama associated with tile (n, m), for $n = 1, \ldots, N$, and $m = 1, \ldots, M$. Now, let $S_j^{nm,V_c} = S_j^{V_c} \cap S_j^{nm}$ denote the set of pixels in tile (n, m) present in the user viewport at that time instance. That is, S_j^{nm,V_c} represents the spatial area in the 360° panorama shared by tile (n, m) and V_c at time t_j.

We illustrate later that a user viewport may occupy very different, in terms of shape and size, spatial areas of the 360° panorama, depending on its latitude (the polar angle θ on the 360° view sphere). To account for this, in developing our statistical model of user navigation, we formulate next the fractions of the spatial areas of every tile, present in the user viewport V_c at t_j, as follows:

$$w_j^{nm} = \frac{|S_j^{nm,V_c}|}{\sum_{n,m} |S_j^{nm,V_c}|}, \tag{1}$$

where $|S|$ denotes the size of a set S, in this case in number of pixels. Thus, $\{w_j^{nm}\}$ represents the normalized distribution of the spatial area of the user viewport across every tile in the 360° panorama, at time instance t_j.

Given (1), we can formulate the probability (likelihood) of the user navigating tile (n, m) over a time interval spanned by the time instances $[t_i, t_j]$, as follows:

$$P_{nm}^{(t_i,t_j)} = \frac{\sum_{k=i}^{j} w_k^{nm}}{j - i + 1}. \tag{2}$$

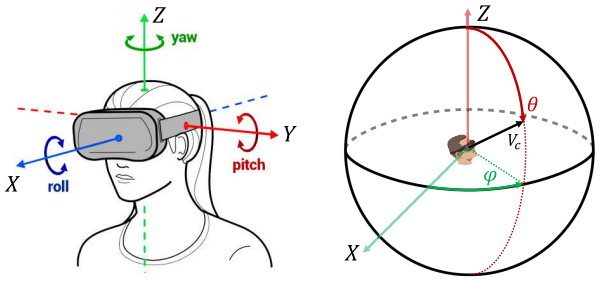

Fig. 4. 360° navigation data of current viewport V_c. Left: Rotation angles yaw, pitch, and roll around the three coordinate axis. Right: Azimuthal and polar angles (φ, θ) in spherical coordinates.

In other words, $P_{nm}^{(t_i,t_j)}$ indicates how often tile (n, m) appears (at least in part) in the user viewport during navigation of the 360° video from its temporal instance t_i to t_j, or the popularity of the 360° scene content captured by the tile for this user and time interval. For instance, if t_i and t_j correspond to the first and last video frame of the 360° video, then, $P_{nm}^{(t_i,t_j)}$ captures the navigation probability or popularity of tile (n, m) across the entire video.

D. Viewport-Adaptive Space-Time Scalability

To enable an effective allocation of system resources across a 360° video, we explore a scalable multi-layer tiling of a 360° panorama. In particular, for every tile (n, m) in the panorama, we will construct L embedded layers of progressively increasing signal fidelity. The multi-layer tiling construction is illustrated in Figure 5. It can enable carrying out effective trade-offs between delivered immersion fidelity and induced data rate, spatiotemporally over the 360° content, in response to the user navigation actions. This can be effectively accomplished by optimally selecting the number of layers l_{nm} sent for every tile (n, m) during a time interval, such that the expected user viewport quality over that interval is maximized, given the available network streaming bandwidth C.

We can formally capture this optimization as:

$$\max_{l_{nm}} \sum_{nm} P_{nm} Q_{nm}(l_{nm}), \quad \text{subject to:} \sum_{nm} R_{nm}(l_{nm}) \leq C.$$

Here, P_{nm} denotes the likelihood of navigating tile t_{nm} during the time period under consideration, as introduced earlier. $Q_{nm}(l_{nm})$ and $R_{nm}(l_{nm})$ denote respectively the delivered immersion fidelity and induced data rate associated with tile t_{nm}, given that its first l_{nm} scalable layers are sent to a user.

Note that the proposed statistical analysis of user navigation captures as navigation likelihoods the expected overlap of a tile with the user viewport over a time interval, and the aspect that equatorial tiles are more likely navigated than polar tiles. Thus, our expected viewport quality formulation can correspond to a tile-level WS-PSNR (Weighted Spherical PSNR) [34].

In our experimental evaluation, we facilitate the scalable extension of HEVC [35], to effectively implement our multi-layer 360° tiling construction from Figure 5.

V. Edge-Based Multi-User VR Streaming

A. System Modeling

We describe here in detail our system modeling associated with the setting we investigate (see Figure 1). There is a set

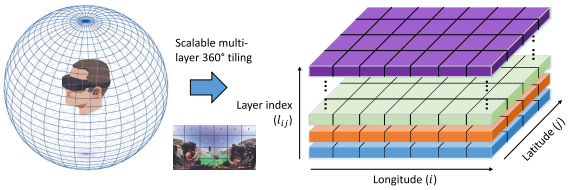

Fig. 5. Scalable multi-layer 360° tiling.

of 360° videos served to mobile VR users at each small-base station. Each 360° video comprises $N \times M$ tiles as introduced earlier. To ease the notation and terminology that we will need to resort to, we assign to every tile (n, m) of an entire 360° video a unique index j and denote it henceforth as video j. We consider that each small base station i serves a set of VR clients that collectively induce a popularity distribution P_{ij} over the tile-videos, by their navigation actions, as introduced earlier. A small base station can store a subset of the videos at its edge server, to deliver them locally to its own users. It can also serve one of its videos to a VR client at another small base station via the backhaul links through which they can cooperate, if this video is not stored locally at the edge server of that small base station. If a requested video is not available locally or from a neighboring small base station, it is delivered remotely from the back-end server.

Let $Y_{ij}^l \in \{0, 1\}$ denote the decision for small base station i to cache tile-video j comprising its first l scalable layers (see Figure 5). Let $X_{ij}^{l,k} \in \{0, 1\}$ denote the decision to deliver video j comprising its first l scalable layers from base station k to a VR user at base station i. If $k = i$, then $X_{ij}^{l,k} = 1$ will denote the event of local delivery at base station i. If k is greater than the number of small base stations in the system, $X_{ij}^{l,k}$ will capture the decision to deliver video j comprising its first l scalable layers remotely, from the back-end server, as introduced earlier. Let $Q_{j,l}$ denote the delivered immersion fidelity of tile-video j comprising its first l scalable layers.

Let $C_{ij}^{l,k}$ denote the cost of delivering tile video j comprising its first l scalable layers to a VR user at small base station i from the cache of small base station k. $C_{ij}^{l,k}$ captures the impact of the relative distance between base stations i and k, and the data volume B_j^l of tile-video j featuring l scalable layers. Similarly, let $\bar{B}_j^l$ denote the processing and rendering cost associated with tile-video j featuring l scalable layers, induced at a small base station. Let Z_i and $\bar{Z}_i$ denote respectively the storage and computing capabilities of the edge server at base station i. In our experimental evaluation, we show that $\bar{B}_j^l$ can be modeled as a polynomial function of the data volume of tile-video j comprising l scalable layers.

We consider the possibility of having the data packets associated with tile-video j encoded using network coding or fountain codes [36], [37]. We explore the implications of this option for our problem formulation and its optimization solution in the next section. In essence, working with such packets helps use system resources more efficiently and reduce the system's transmission scheduling complexity. These advances stem from their construction which eliminates packet duplication and thus enables working with fractional network flows, instead of their discrete $\{0, 1\}$ counterparts.

The method for constructing network coding packets we pursue is illustrated in Figure 6. The bitstream representing the

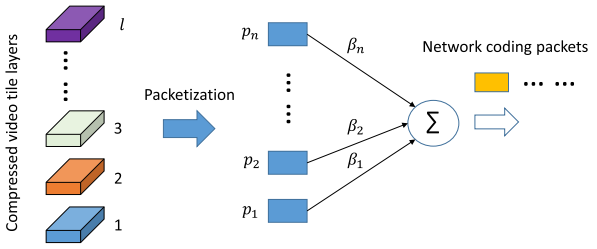

Fig. 6. Video tile layers to data packets to network coding packets.

TABLE I
MAJOR SYMBOLS USED IN THE SYSTEM MODELING

Symbol	Description
P_{ij}	Popularity of tile-video j at small base station (SBS) i
Y_{ij}^{l}	Cache l layers of tile-video j at SBS i
$X_{ij}^{l,k}$	Stream l layers of tile-video j from SBS k to user at SBS i
$Q_{j,l}$	Immersion fidelity of l layers of tile-video j
$C_{ij}^{l,k}$	Cost to stream l layers of tile-video j from SBS k to SBS i
B_j^l	Data volume of l layers of tile-video j
$\bar{B}_j^l$	Processing and rendering cost of l layers of tile-video j
$Z_i, \bar{Z}_i$	Storage and computing capabilities of edge server at SBS i

scalable compressed video tile layers is first packetized into data packets p_i. Network coding packets are then constructed as weighted linear combinations of the data packets using $\sum_i \beta_i p_i$, where the weights β_i are selected uniformly at random from a Galois finite field. The arithmetic operation of summation is performed over the same finite field.

Table I summarizes the main notation described herein.

B. Problem Formulation

We are interested in maximizing the immersion fidelity delivered to the VR clients at the small base stations, while minimizing the induced cost. We analytically formulate this problem of interest as follows. Given a tile-video j featuring l scalable layers and a small base station i, let $Q_{j,l}/C_{i,j}^{l,k}$ denote the delivered immersion fidelity per unit cost, when this content is delivered from small base station k. We recall that in this case the decision variables Y_{kj}^{l} and $X_{ij}^{l,k}$ would need to be set to one. Now, let $\sum_{i,j,k,l} P_{ij} X_{ij}^{l,k} Q_{j,l}/C_{i,j}^{l,k}$ denote the aggregate expected delivered immersion fidelity per unit cost. Our objective is to maximize this quantity given various system and problem formulation constraints that arise here.

We formally characterize our objective as:

$$\max_{Y_{ij}^{l}, X_{ij}^{l,k}} \sum_{i,j,k,l} P_{ij} X_{ij}^{l,k} Q_{j,l}/C_{i,j}^{l,k}, \tag{3}$$

$$\text{s.t.: } X_{ij}^{l,k} \leq Y_{kj}^{l}, \quad \forall i, j, k, l, \sum_{l} Y_{ij}^{l} \leq 1, \ \forall i, j, \tag{4}$$

$$\sum_{k} X_{ij}^{l,k} \leq 1, \quad \forall i, j, l, \sum_{j,l,k} X_{ij}^{l,k} B_j^l \leq Z_{bh}, \ \forall i, \tag{5}$$

$$\sum_{j,l} Y_{ij}^{l} B_j^l \leq Z_i, \quad \forall i, \sum_{j,l} Y_{ij}^{l} \bar{B}_j^l \leq \bar{Z}_i, \ \forall i, \tag{6}$$

where the first constraint in (4) captures the notion that tile-video j cannot be delivered from small base station k, unless it is cached there. The second constraint in (4) captures the condition that only one replica of tile-video j comprising l scalable layers is stored at base station i. The first constraint in (5) captures the notion that tile-video j comprising l scalable layers is streamed to VR clients at small base station i from the edge server of only one small base station k. The second constraint in (5) ensures that the tile-video data streamed to any small base station i does not exceed the transmission capacity of the carrier backhaul links, denoted here as Z_{bh}. Finally, the two constraints in (6) capture the limited caching and computing capabilities of the edge servers at every small base station, as introduced earlier.

C. Analysis and Approximation

The problem (3) - (6) is discrete and has a combinatorial nature. Thus, it is difficult to solve. We first show that this problem is NP-complete. We then formulate a polynomial-time approximation solution via dynamic programming. Finally, we analyze the quality of the resulting approximation. The analysis builds upon our preliminary work in [30] and addresses the additional challenges introduced by the integration of scalable 360° tiled video data, limited edge computing capabilities, and a more general problem formulation, explored in the present paper. We conclude this section by analyzing the impact of using network coding packets on the problem formulation (3) - (6) and the resulting optimization methods that it will require.

Showing that (3) - (6) is NP-complete requires showing that any given solution can be verified quickly and that a known NP-complete problem can be reduced to (3) - (6) [38]. We can verify a given solution by checking its feasibility against the constraints (4) - (6) in polynomial time. This meets the first requirement. We meet the second requirement by mapping the known NP-complete multi-knapsack problem [21] to (3) - (6).

The multiple knapsack problem comprises N items, characterized with profit and weight factors α_n and γ_n, and K knapsacks, characterized with holding capacity factors c_k. The objective is to select K disjoint subsets of items such that their aggregate profit is maximized and each can be assigned to a knapsack k such that its aggregate weight does not exceed c_k. We extend this definition to include two weight factors per item n, namely γ_n^1 γ_n^2, and respectively two capacity factors c_k^1 and c_k^2 per knapsack k. The NP-complete nature of the problem remains under the extension. We map this problem then to (3) - (6) as follows. First, we map each knapsack to the edge server associated with one small base station, such that $Z_k = c_k^1$ and $\bar{Z}_k = c_k^2$. Next, we map item n to tile video j comprising l scalable layer such that $B_j^l = \gamma_n^1$ and $\bar{B}_j^l = \gamma_n^2$, and $\sum_i P_{ij} = \alpha_n$. Finally, we set $Q_{j,l}/C_{i,j}^{l,k} = 1$, $\forall i, j, l, k$.

Given the above, the knapsack problem and the mapped instance of our problem share a feasible solution with a common objective function value. Moreover, we can carry out the problem mapping reduction above in polynomial time. This completes the verification that (3) - (6) is NP-complete.

To formulate an approximation solution to solve (3) - (6), we note that given our system setting in Figure 1, the cost factors $C_{i,j}^{l,k}$ will all be equal when streaming tile video j featuring l scalable layers from any neighboring small base station $k \neq i$, and smaller than the cost factor of delivering this content from the remote back-end server. Thus, we can

rewrite (3) - (6) as:

$$\max_{Y_{in}, X^k_{in}} \sum_{i,n} P_{in} X^i_{in} Q_n / C^i_{i,n} + \sum_{i,n,k\neq i} P_{in} X^k_{in} Q_n / \bar{C}_n,$$
$$\text{subject to: } (4) - (6). \tag{7}$$

To simplify the notation, we have mapped each original index pair (j, l) to a unique single variable n in the reformulation above. $\bar{C}_n$ denotes the common cost factor of delivering content item n from any neighboring small base station $k \neq i$.

Let $\alpha_{in} = P_{in} Q_n / C^i_{i,n} + \sum_{k\neq i} P_{kn} Q_n / \bar{C}_n$ denote the maximum prospective benefit of caching item n at small base station i. We define $\mathcal{V} = \{1, \ldots, K\} \times \{1, \ldots, N\}$ to be the set representing the vector product of the sets of small base stations and content items, where K and N denote their respective sizes. Let $\upsilon = (i, n) \in \mathcal{V}$ denote a member of this set. Facilitating $\mathcal{V}$, we can solve (7) as a multiple knapsack problem with multiple constraints associated with each small base station k, as follows.

Using dynamic programming [39], we formulate the optimal value function $f_\upsilon(\cdot)$ associated with (7) as

$$f_\upsilon(s_1, \bar{s}_1, \ldots, s_K, \bar{s}_K) = \begin{cases} \max_{x_\upsilon \in \{0,1\}} \{\alpha_{in} x_\upsilon + f_{\upsilon-1}(\ldots, s_i - B^l_j x_\upsilon, \bar{s}_i - \bar{B}^l_j x_\upsilon, \ldots)\}, \\ \quad \text{if } \nexists \upsilon' < \upsilon : x_{\upsilon'} = 1 \wedge n' = n \wedge s_i \geq B^l_j, \bar{s}_i \geq \bar{B}^l_j, \\ \max_{x_\upsilon \in \{0,1\}} \{(\alpha_{in} - P_{in} Q_n / \bar{C}_n) x_\upsilon \\ \qquad + f_{\upsilon-1}(\ldots, s_i - B^l_j x_\upsilon, \bar{s}_i - \bar{B}^l_j x_\upsilon, \ldots)\}, \\ \quad \text{if } \exists \upsilon' < \upsilon : x_{\upsilon'} = 1 \wedge n' = n, i' \neq i \wedge s_i \geq B^l_j, \bar{s}_i \geq \bar{B}^l_j, \\ f_{\upsilon-1}(s_1, \bar{s}_1, \ldots, s_K, \bar{s}_K), \\ \quad \text{if } \exists \upsilon' < \upsilon : x_{\upsilon'} = 1 \wedge n' = n, i' = i \vee s_i < B^l_j \vee \bar{s}_i < \bar{B}^l_j, \end{cases}$$

for $\upsilon = |\mathcal{V}|, \ldots, 1$, where $f_0(\cdot) = 0$. The state variables $s_i \in \{0, \ldots, Z_i\}$ and $\bar{s}_i \in \{0, \ldots, \bar{Z}_i\}$ capture the slack caching and computing capacity, respectively, at small base station i.

We develop the optimal value function $f_\upsilon(\cdot)$ using the above Bellman optimality condition recursion, in stages, iteratively, starting from stage $\upsilon = 1, \ldots, |\mathcal{V}|$. Our objective is to determine $f_{|\mathcal{V}|}(Z_1, \bar{Z}_1, \ldots, Z_K, \bar{Z}_K)$, which corresponds to the objective in (7) at the optimal solution $\{x^*_\upsilon\}$. The latter can then be obtained by backtracking from $f_{|\mathcal{V}|}(Z_1, \bar{Z}_1, \ldots, Z_K, \bar{Z}_K)$.

Completing $f_\upsilon(\cdot)$ requires a total running time of $O(|\mathcal{V}|KM)$, where $M = \max_i\{Z_i, \bar{Z}_i\}$. Thus, this approach represents a pseudo-polynomial time algorithm for solving (7).

We proceed one step further to formulate a fully-polynomial time approximation scheme [38] for solving (7) that will leverage the above development. The formulation will integrate an efficient branch-and-prune methodology of keeping track of only the optimal paths in a data tree structure capturing the different stages of growing the optimal value function $f_\upsilon(\cdot)$. We conclude with a verification of the fully-polynomial time nature of the formulated approximation scheme.

In particular, we first scale the benefit factors α_{in} associated with caching item n at small base station i, such that they are all small numbers, polynomially bounded in $|\mathcal{V}|$. Applying dynamic programming via the optimal value function, as formulated above, to the scaled instance of (7) would then result in a polynomial running time (in $|\mathcal{V}|$) solution strategy, with an induced approximation factor ϵ. Moreover, we integrate a desired ϵ into the scaling of the factors α_{in}, so that this strategy is fully-polynomial time, i.e., with respect to $1/\epsilon$, as well.

Algorithm 1 Branch-and-Prune Approximation Scheme

Initialize $\mathbf{T} = \emptyset, F_0 = 0, \mathbf{Q}_0 = \{(\mathbf{T}, F_0)\}$
for $\forall \upsilon \in \mathcal{V}$ **do**
  `Expansion Phase`
  for $\forall (\mathbf{T}, P_{\upsilon-1}) \in \mathbf{Q}_{\upsilon-1}$ **do**
    Branch $\mathbf{Q}_\upsilon = \mathbf{Q}_{\upsilon-1} \cup \{(\mathbf{T} \cup \{\upsilon\}, F_{\upsilon-1} + \alpha_\upsilon)\}$
    Subject to:
      (i) $\sum_{\upsilon' \in \mathbf{T}} B_{n'} + B_n \leq Z_i \wedge \sum_{\upsilon' \in \mathbf{T}} \bar{B}_{n'} + \bar{B}_n \leq \bar{Z}_i$
      (ii) If $\nexists \upsilon' \in \mathbf{T} : n' = n$,
        Set $\alpha_\upsilon = \alpha^s_{in}$
      (iii) If $\exists \upsilon' \in \mathbf{T} : n' = n \wedge i' \neq i$,
        Set $\alpha_\upsilon = \lfloor \frac{P_{in} Q_n / C^i_{i,n}}{10^p} \rfloor - \lfloor \frac{P_{kn} Q_n / \bar{C}_n}{10^p} \rfloor$
  end for
  `Prunning Phase`
  if $\exists (\mathbf{T}', F'_\upsilon), (\mathbf{T}'', F''_\upsilon) \in \mathbf{Q}_\upsilon$
    Subject to:
    (i) $F'_\upsilon = F''_\upsilon$, and
    (ii) $\sum_{\upsilon \in \mathbf{T}'} B_n > \sum_{\upsilon \in \mathbf{T}''} B_n \wedge \sum_{\upsilon \in \mathbf{T}'} \bar{B}_n \geq \sum_{\upsilon \in \mathbf{T}''} \bar{B}_n$, or
    (iii) $\sum_{\upsilon \in \mathbf{T}'} B_n \geq \sum_{\upsilon \in \mathbf{T}''} B_n \wedge \sum_{\upsilon \in \mathbf{T}'} \bar{B}_n > \sum_{\upsilon \in \mathbf{T}''} \bar{B}_n$,
    then prune $\mathbf{Q}_\upsilon = \mathbf{Q}_\upsilon \setminus \{(\mathbf{T}', F'_\upsilon)\}$
  end if
end for
Select $(\mathbf{T}^*, F^*_{|\mathcal{V}|}) : F^*_{|\mathcal{V}|} = \arg\max_{F_\upsilon}\{(\mathbf{T}, F_{|\mathcal{V}|}) \in \mathbf{Q}_{|\mathcal{V}|}\}$

Let $\alpha^{\max} = \max_{i,n} \alpha_{in}$ denote a scaling factor we will use. Let $p = \lfloor \log(\epsilon \alpha^{\max} / |\mathcal{V}|) \rfloor$ capture the precision at which we will approximate/quantize the benefit factors α_{in}. Then, we define $\alpha^s_{in} = \lfloor \frac{P_{in} Q_n / C^i_{i,n}}{10^p} \rfloor + \sum_{k\neq i} \lfloor \frac{P_{kn} Q_n / \bar{C}_n}{10^p} \rfloor$.

Algorithm 1 outlines our efficient branch-and-prune algorithmic implementation of the above strategy, as noted earlier.

In particular, Algorithm 1 leverages an efficient tree data structure $\mathbf{Q}_\upsilon$ that is dynamically updated during execution. Member elements $(\mathbf{T}, F_\upsilon) \in \mathbf{Q}_\upsilon$ comprise subsets $\mathbf{T}$ of size $\leq \upsilon$, of the first υ elements in $\mathcal{V}$, such that they induce the maximum achieved benefit (F_υ), given the constraints (4) - (6). For every subsequent υ, Algorithm 1 comprises *an expansion phase*, where the optimal paths (subsets of cached data) maintained in $\mathbf{Q}_{\upsilon-1}$ are branched out by considering the next decision variable υ (to cache item n at base station i), while observing constraints (4) - (6), and *a pruning phase*, where only the optimal paths after the expansion are retained.

At completion of stage $\upsilon = |\mathcal{V}|$, Algorithm 1 terminates by selecting the caching configuration $\mathbf{T}^*$ in $\mathbf{Q}_{|\mathcal{V}|}$ that exhibits the maximum achieved benefit $F^*_{|\mathcal{V}|}$. Then, the corresponding optimal streaming variables can be selected as follows. First, $\forall \upsilon \in \mathbf{T}^*$, we set $X^i_{in} = 1$. Next, if $\exists i, k \neq i : X^i_{in} = 1 \wedge X^m_{kn} = 0, \forall m$, we set $X^i_{kn} = 1$. Last, if $\exists i \ : X^k_{in} = 0, \forall k$, we set $X^{K+1}_{in} = 1$.

Finally, we verify the approximation guarantees of Algorithm 1. Let OPT denote the optimal objective in (7) and let $\{X^k_{in}\}^*$ denote the respective solution. We want to verify that the solution $\{X^k_{in}\}'$ computed by Algorithm 1 satisfies $O(\{X^k_{in}\}') \geq (1 - \epsilon) \cdot \text{OPT}$, where $O(\cdot)$ denotes the objective

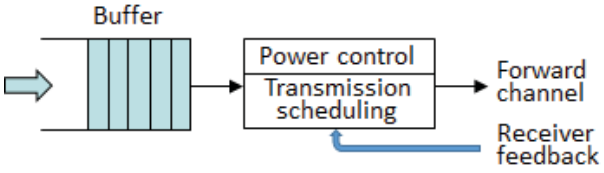

Fig. 7. A small base station's packet transmission system model.

function in (7). In particular, Algorithm 1 operates on scaled benefit factors, where $\Delta = \epsilon\alpha^{\max}/|\mathcal{V}|$ denotes the scaling aspect. Thus, the benefit α^s_{in} achieved by selecting item v in the solution $\{X^k_{in}\}'$ will satisfy $\Delta\alpha^s_{in} \leq \alpha_{in}$. This implies that the achieved benefit induced by $\{X^k_{in}\}'$ can drop at most Δ, for every item v cached according to $\{X^k_{in}\}^*$. Hence, we can bound the overall achieved benefit drop as $O(\{X^k_{in}\}^*) - \Delta \cdot O'(\{X^k_{in}\}^*) \leq |\mathcal{V}|\Delta$. Here, O' denotes the objective function in (7) evaluated on the scaled benefit factors.

On the other hand, the solution $\{X^k_{in}\}'$ computed by Algorithm 1 represents the optimal solution for the scaled instance of the problem (7), (4) - (6). Thus, $O'(\{X^k_{in}\}') \geq O'(\{X^k_{in}\}^*)$. Leveraging these two inequality relationships, we can write

$$\begin{aligned} O(\{X^k_{in}\}') \geq \Delta \cdot O'(\{X^k_{in}\}') &\geq \Delta \cdot O'(\{X^k_{in}\}^*) \\ &\geq O(\{X^k_{in}\}^*) - |\mathcal{V}|\Delta \quad (8) \\ &= \text{OPT} - \epsilon\alpha^{\max} \\ &\geq (1-\epsilon) \cdot \text{OPT} \quad (9) \end{aligned}$$

where (8) follows from the first inequality relationship established earlier and (9) holds as $\text{OPT} \geq \alpha^{\max}$. This verifies the desired approximation guarantees of Algorithm 1.

The running time of Algorithm 1 is polynomial in $|\mathcal{V}|$, as it corresponds to completing a table of at most $|\mathcal{V}|^2\lfloor\alpha^{\max}/\Delta\rfloor$ entries. The scaling enables the running time of Algorithm 1 also to be polynomial in $1/\epsilon$, as $|\mathcal{V}|\alpha^{\max}/\Delta = 1/\epsilon$.

D. Streaming Network Coding Packets

Using network coding packets reduces the complexity of (3) - (6), as the decision variables Y^l_{kj} and $X^{l,k}_{ij}$ can be continuous in that case. Thus, (3) - (6) becomes linear programming, which can be solved exactly in polynomial time [40].

In particular, relaxing $Y^l_{kj}, X^{l,k}_{ij} \in [0,1]$ to be fractional will capture that now portions of the network coding packets representing tile-video j featuring l scalable layers can be cached at small base station k, and streamed from this base station to users at small base station i, respectively. Hence, the objective (3) will become a linear weighted-sum function of continuous variables indicating the proportions of network coding packets associated with a specific tile-video streamed to users at a given small base station, from each base station. Moreover, the constraints (4) - (6) will become linear functions as well and will still hold after relaxing the original discrete decision variables. Specifically, all constraints from (4) - (6) will apply in a straightforward manner, with for instance $Y^l_{kj}B^l_j$ and $X^{l,k}_{ij}B^l_j$ indicating in this case the data volumes associated with the proportions of network coding packets representing tile-video j comprising l scalable layers cached at small base station k, and streamed from this small base station to users at base station i, respectively.

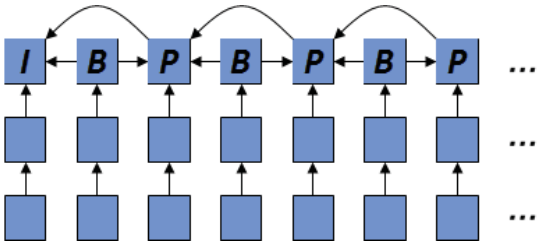

Fig. 8. Acyclic graph of scalable 360° video packet dependencies.

We note that the constraint $\sum_k X^{l,k}_{ij} \leq 1$ in (5) requires that tile-video j comprising l scalable layers be streamed to users at small base station i, from at most one base station k, in the original case of discrete decision variables, to avoid duplicate delivery. Network coding packets ensure the latter by design,[1] thus requiring only for the same number packets to be delivered on aggregate from across the entire set of small base stations. That is, complementary subsets of packets can be streamed from each base station k in parallel, thereby making the streaming scheduling more efficient, with $X^{l,k}_{ij}$ indicating the fraction each subset represents of the entire whole.

VI. Base Station 360° Packet Scheduling

We explore rate-distortion-power optimized scheduling of buffered packets at the base stations to overcome network transients. In particular, a base station may experience transient periods during which its actual transmission rate capacity may be temporarily lower than its incoming rate of outbound packets. This will necessitate buffering such packets and effective scheduling of their outgoing transmissions. We will explore this challenge via the setup illustrated in Figure 7.

We consider there are L packets in the transmission buffer of a base station. The packets feature dependencies induced upon them at encoding that can be characterized as an acyclic graph, as illustrated in Figure 8. In particular, $i \to j$ (a directed edge from node i to node j in the graph) indicates that packet j is required to decode packet i. Similarly, $j \preceq l$ denotes all ancestors of data unit l in the coding hierarchy. We characterize each packet l with a delivery deadline $t_{l,d}$, size B_l in bytes, and reduction in reconstruction error ΔD_l of the respective 360° video that l will contribute to, if received/decoded on time.

We characterize the forward/downlink channel with a packet erasure probability $\epsilon \triangleq \epsilon(h,c)$, where h denotes the current channel state/quality, as informed by receiver feedback, and c the selected transmit power, and transmission delay τ of density $p_\tau \triangleq p_\tau(h,c)$. Let $\pi = (\pi_1, \ldots, \pi_L)$ be the packet transmission policy of the station, where $\pi_l \in \{0,1\}$ indicates the two possible choices of not sending or sending packet l at present (current time t). Let $c = (c_1, \ldots, c_L)$ be its transmit power policy. Finally, let $\epsilon(\pi_l, c_l) \triangleq P\{\tau_l > t_{l,d} + \alpha - t\}$ be the expected error of transmitting packet l under policy π_l, where α captures the latency/interactivity requirements of the VR application. We can formulate it as:

$$\epsilon(\pi_l, c_l) = \begin{cases} 1, & \text{if } \pi_l = 0, \\ \epsilon_l + (1-\epsilon)\int_{t_{l,d}+\alpha-t}^{\infty} p_\tau, & \text{if } \pi_l = 1. \end{cases} \quad (10)$$

Next, we formulate the problem of interest. Let $D(\pi, c) = \sum_l \Delta D_l \prod_{j \preceq l} \left(1 - \epsilon(\pi_j, c_j)\right)$ denote the aggregate expected

[1] Selecting the Galois field size to be large ensures linear independence of the generated network coding packets [37].

Algorithm 2 Optimal Transmission Scheduling Policy

Require: $\pi_l^{(0)}, c_l^{(0)}, J^{(0)}, \lambda_1, \lambda_2, n = 0, \theta$
1: **repeat**
2: $n = n + 1; l = (n \mod L)$
3: Solve (14) $\Rightarrow \pi_l^{(n)}, c_l^{(n)}$
4: **until** *convergence* $(J^{(n-1)} - J^{(n)} < \theta)$

reconstruction error reduction over the 360° tile-videos associated with the buffered packets, given (π, c). We formulate the respective transmit rate/power induced by them as $R_T(\pi) = \sum_l \pi_l B_l$ and $E(\pi, c) = \sum_l c_l \pi_l B_l$. We investigate the best scheduling policy (π, c) via the optimization

$$\max_{\pi,c} D(\pi, c), \quad \text{subject to: } R_T(\pi) \leq R_i^*, \ E(\pi, c) \leq E_B, \quad (11)$$

where E_B is the transmit power budget and C is the downlink transmit rate capacity. We explore exact and approximate lower-complexity techniques to solve (11).

In particular, facilitating the Generalized Lagrange multiplier method [41], we reformulate (11) as

$$\min_{\pi,c} J(\pi, c) = -D(\pi, c) + \lambda_1 R_T(\pi) + \lambda_2 E(\pi, c), \quad (12)$$

where $\lambda_i > 0$ denote the corresponding Lagrange multipliers. Moreover, for mathematical convenience, we introduce a minus sign in front of $D(\pi, c)$ and replace the max operator from (11) with a min operator. We then compute the optimal policy via the Bellman optimality condition [39]:

$$(\pi^*, c^*)(q_i) = \arg\min_{\pi_i, c_i} \sum_{q_{i+1}} P_{\pi,c}(q_{i+1}|q_i) J_{\pi^*,c^*}(q_{i+1}), \quad (13)$$

where q_i is a state in the joint policy space of (π, c), uniquely captured by the actions $\pi_1^*, \ldots, \pi_i^*$ and $c_1^*, \ldots, c_i^*$. $P_{\pi,c}(q_{i+1}|q_i)$ are state transition probabilities induced by (π, c), and $J_{\pi^*,c^*}(q_{i+1})$ is the optimal Lagrange cost that we can backtrack with an equivalent equation.

Moreover, we study minimizing $J(\pi, c)$ iteratively, one policy pair (π_l, c_l) at a time. We note that (11) represents a discrete optimization problem that is complex to solve, due to its large state-space that requires an enumeration of the $2^L \times |C|^L$ choices that (π, c) can take on jointly. Thus, we also design a faster iterative algorithm that computes an approximate solution at lower complexity, as follows. Starting from an initial solution for (π, c), we iteratively solve (12) one variable pair (π_l, c_l) at a time, while keeping the others $((\pi_j, c_j)$, for $j \neq l)$ fixed, until convergence.

Concretely, let $(\pi^{(0)}, c^{(0)})$ indicate an initial choice for the joint transmission scheduling and power control policy. Let $n = 1, 2, \ldots$ represent the iteration count. We select one policy pair $(\pi_l^{(n)}, c_l^{(n)})$ to optimize at iteration n, e.g., in a round-robin fashion, for $l = 1, \ldots, L$. We fix the remaining policy pairs $(\pi_j^{(n)}, c_j^{(n)}) = (\pi_j^{(n-1)}, c_j^{(n-1)})$, for $j \neq l$, and compute the values of $(\pi_l^{(n)}, c_l^{(n)})$ that solve (12). We then increment n and move on to the next l, until $J(\pi^{(n)}, c^{(n)}) = J(\pi^{(n-1)}, c^{(n-1)})$.

By grouping terms in (12) associated with (π_l, c_l), we can formulate the key step of the iterative optimization as

$$\pi_l^{(n)}, c_l^{(n)} = \arg\min_{\pi_l, c_l} S_l^{(n)} \epsilon(\pi_l) + \lambda \pi_l B_l, \quad (14)$$

where $\lambda = \lambda_1 + \lambda_2 \ c$ and $S_l^{(n)}$ captures the overall impact of packet l on the reconstruction error of its 360° video. Similarly, we derive $S_l^{(n)}$ from (12) as

$$S_l^{(n)} = \sum_{j \geq l} \Delta D_j \prod_{\substack{m \leq j \\ m \neq l}} (1 - \epsilon(\pi_m^{(n)})).$$

We formally summarize the optimization in Algorithm 2. Its convergence is guaranteed, as the objective function $J(\pi, c)$ is bounded from below and is monotonically non-increasing at every iteration.

VII. Experiments

A. Experimentation Methodology and Data Characteristics

To evaluate the performance of our system framework, we carry out comprehensive simulation experiments that investigate diverse performance aspects of the several major components comprising it. To assist the experimental analysis, we have implemented the scenario under consideration from Figure 1 as follows. There are three small base stations and five users assigned to each base station. The users are distributed uniformly at random across the spatial area served by the respective small cell. There are eight 360° videos available in the system that the users can request for streaming. Following earlier studies, we consider that the likelihood that a user requests a specific video file can be modeled using a Zipf distribution with a parameter γ set to 0.8 [42]. The 360° video content comprises popular sequences used in earlier studies that feature diverse characteristics [14], [43]. These are Coaster, Wingsuit, Paris, Basketball, Runner, Angel Falls, Kite Flite, and Dolphins. They feature spatial 360° panorama resolutions of 4K or 8K, and temporal frame rates of 30 or 60 fps.

We have implemented our novel multi-layer 360° tiling proposed in Section IV using the scalable extension of the latest video compression standard known as SHVC [35], to construct the 360° content into $L = 3$ scalable layers, Group of Pictures (GOP) size of 30 frames, and 6×4 spatial tiles. These advances enabled us to collect the immersion fidelity factors $Q_{j,l}$ introduced in Section V. All reference methods examined in our evaluation use instead 360° video content corresponding to the above sequences, compressed using HEVC into GOPs of 30 frames, as in our case. We leveraged extensive 360° video user navigation traces that we have captured as part of our ongoing work and data collection campaign [14], [15], [44], [45], to enable our statistical models of user video tile navigation formulated in Section IV-C.

We related the cost factors $C_{i,j}^{l,k}$ introduced in Section V to the energy consumption of streaming the 360° content. Concretely, if l scalable layers of tile-video j are cached locally at small base station i, the energy consumption cost will reflect the base station transmission only. If they are retrieved from another base station k, the consumed energy will represent the sum of the energy spent to move the content between i and k, and the energy spent by i to transmit them. We considered that the energy required to move content between small base stations is proportional to the number of network hops between them (in our case one). We alike accounted for the cost of streaming l scalable layers of tile-video j from the back-end server, assuming that the data will traverse 15 hops in this case,

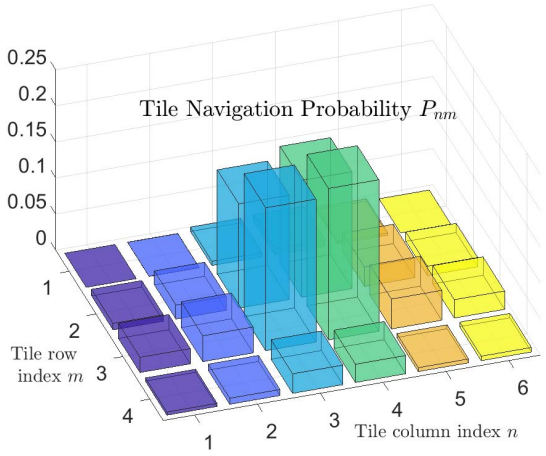

Fig. 9. Navigation probabilities for 360° tiles for Roller Coaster.

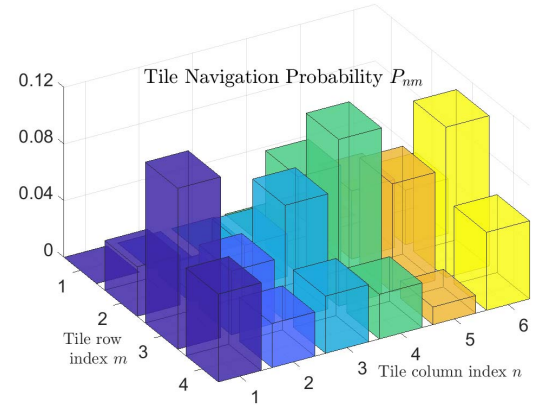

Fig. 10. Navigation probabilities for 360° video tiles for Wingsuit.

based on [46]. Finally, following [47], [48], we considered that the energy per bit consumed by wireless transmission is 3.5 μJ and that for wireline transmission is 0.5 μJ per hop.

B. Reference Methods

We have implemented several state-of-the-art reference methods, to benchmark the performance of our system framework and its major components against them, as part of our evaluation. To examine the pure streaming performance of our scalable multi-layer 360° tiling from Section IV and the associated viewport-adaptive allocation of resources suggested therein, we have implemented the state-of-the-art video streaming standard MPEG-DASH [6], [31]. We examine the trade-off between delivered 360° viewport quality and available streaming data rate enabled by these two approaches.

We have also implemented the following two caching techniques, *Blasco2014* [20] and *LRU* (Least-Recently-Used), to compare against our cooperative edge-based multi-user VR streaming and the associated Algorithm 1 for optimal allocation of storage and computing resources therein, proposed in Section V. As introduced earlier, *Blasco2014* implements a caching policy that aims to minimize the aggregate delay of content retrieval at a base station. Moreover, *LRU* is a very popular method used in practice, which implements a caching policy that evicts the least recently used cached items first, when updating the stored content.

Finally, we have implemented two state-of-the-art methods for transmission error control, to benchmark the efficiency of our 360° packet scheduling proposed in Section VI to address transient packet buffering at the base stations. These are hybrid ARQ that combines Automatic Repeat reQuest (ARQ) scheduling with efficient channel coding via Reed-Solomon codes [49], to help address more effectively packet loss during transmission, and the method proposed in [37] that introduces unequal error protection in video delivery via network coding. We denote the former reference method as *HARQ* and the latter reference method as *Thomos2011*.

Fig. 11. Viewport at $(\varphi, \theta) = (0°, 0°)$.

Fig. 12. Viewport at $(\varphi, \theta) = (120°, -60°)$.

C. Ablation Evaluation

360° user viewport tile navigation probabilities. We have constructed the statistical models of user navigation following our approach proposed in Section IV-C. Here, we illustrate the induced tile navigation probabilities for two video sequences used in our evaluation. In particular, in Figure 9 and Figure 10 we show $P_{nm}^{(t_i,t_j)}$ defined in Section IV-C, for t_i and t_j corresponding to the first and last video frame of a 360° video, for Roller Coaster and Wingsuit, respectively.

We can observe from Figure 9 that in the case of Roller Coaster, video tiles on the fringes of the 360° panorama are rarely navigated by a user, i.e., they scarcely appear in the user's viewport, during a 360° video navigation session. Conversely, video tiles indexed as $(n, m) = (3,2)$, $(4,2)$, $(3,3)$, and $(4,3)$ are quite often navigated by the user, as noted by their much higher navigation likelihoods shown in Figure 9.

We can observe from Figure 10 that the navigation probabilities of 360° video tiles for the sequence Wingsuit, induced by users experiencing the content, look fairly different. This is due to the nature of this content and the induced specific interests of the users, expressed when navigating it. In particular, we can see now that the typical user is predominantly interested in navigating the southern hemisphere of the 360° panorama, as seen from the respective tile navigation probabilities indicated by Figure 10 therein. The more dynamic and interesting content of Wingsuit resides spatially in the southern hemisphere of its 360° video sphere.

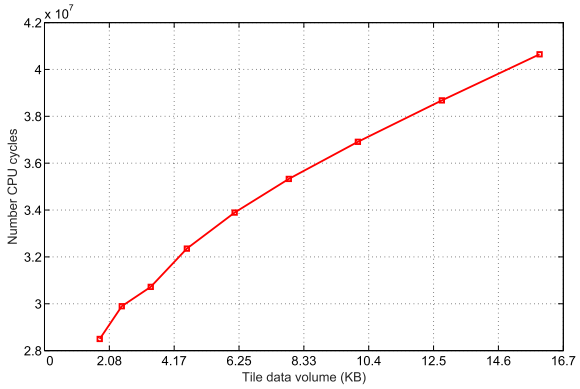

Fig. 13. Required computing power vs. data volume of a 360° tile.

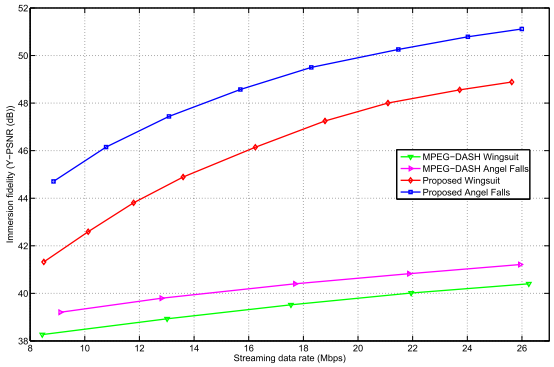

Fig. 14. 360° video transmission efficiency vs. MPEG-DASH.

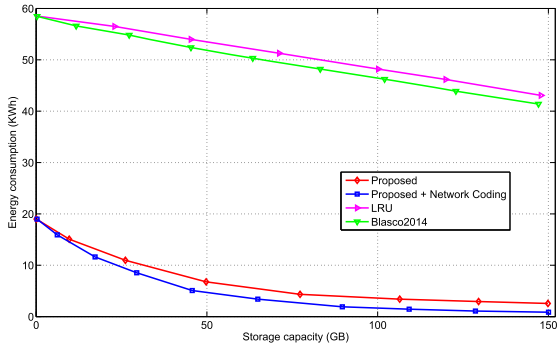

Fig. 15. Energy consumption vs. storage capacity at base stations.

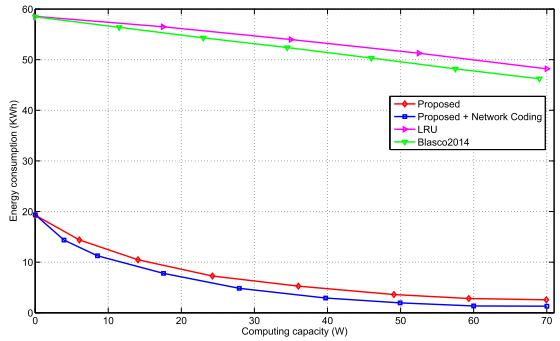

Fig. 16. Energy consumption vs. computing capacity at base stations.

In Section IV-C, we explained the need to normalize the relative impact of the spatial area of a user viewport across the video tiles (n, m) comprising the 360° panorama of a 360° video frame at time instance t_j, as carried out in (1), when formulating our statistical models of 360° user navigation. We noted that this is due to the considerably uneven areas occupied by diverse viewports, depending on their latitude on the 360° sphere. We illustrate this aspect here, by visualizing two such representative viewports in Figure 11 and Figure 12.

Due to applying the equirectangular projection to map the 3D 360° sphere to a wide 2D panorama, as illustrated in Figure 2 and explained in Section II, the shape of a viewport considerably changes. In equatorial regions of the 2D 360° panorama, a projected viewport is smaller and more compact (see Figure 11), while in polar regions a projected viewport is spread over all polar tiles (see Figure 12) and occupies a wider spatial area. Figure 5 can be referenced to understand the spatial locations of these two viewports relative to the underlying tiling of the respective 360° video.

Computing workload vs. tile data volume. To support our optimization techniques, we investigated the computing workload imposed on a microprocessor when decoding compressed video tiles and rendering the corresponding 360° panorama image data they represent. In particular, we empirically characterized the required number of CPU cycles to decompress and render a 360° video tile, as a function of its data volume. These finding are shown in Figure 13, where we can see that the resulting dependency follows closely a polynomial model. We leveraged this dependency to then develop a closed form third order polynomial model that accurately captures the computing workload induced on an edge server as a function of the desired streaming data rate for a 360° video tile. Finally, we integrated this model into our optimization framework, to enable rigorous allocation of the computing resources available at an edge server of a small base station.

Advantages of scalable tiling and user-viewport adaptation. To examine the transmission efficiency of the viewport-adaptive 360° space-time scalability investigated in Section IV-D, which integrates our statistical models of 360° VR user navigation, we carried out the following experiment. A sender transmits 360° content to a VR client over a network link of a given data rate, as highlighted by Figure 2. We implemented the optimization we formulated in Section IV-D to decide how the available network bandwidth C should be allocated across the scalable tile layers associated with the 360° content, in response to the navigation actions of the user, and measured the resulting expected immersion fidelity experienced by the user. We formally measure the latter quantity as the luminance (Y) peak-signal-to-noise ratio (Y-PSNR) of the 360° viewport video signal reconstructed on the user's VR device. This metric has multiple analysis and implementation advantages, and correlates closely with other metrics proposed before [50]. We vary the available network bandwidth C and record the corresponding immersion fidelity delivered to the user, enabled by our approach, in each case. Simultaneously, we recorded the corresponding performance of MPEG-DASH in the same context. Two 360° video sequences from our data set, Wingsuit and Angel Falls, have been used in this analysis.

We show the respective results in Figure 14. We can see that our approach enables considerable benefits over MPEG-DASH by integrating the user navigation actions and the rate-distortion trade-offs across the 360° panorama, in deciding how transmission resources should be allocated.

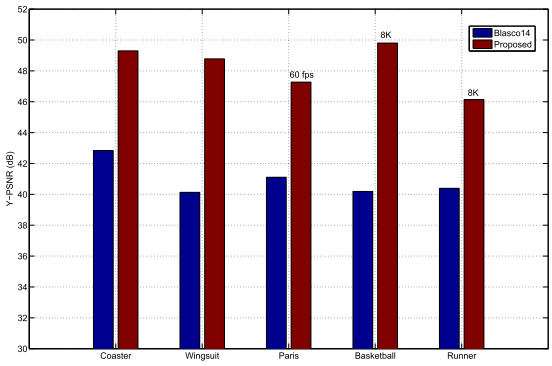

Fig. 17. Delivered immersion fidelity across different 360° videos.

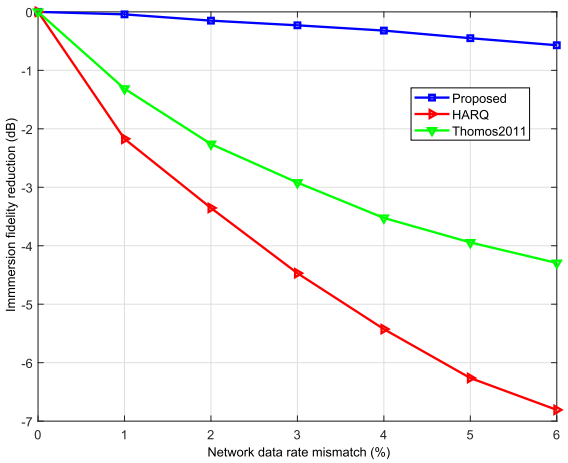

Fig. 18. Immersion fidelity reduction vs. network rate mismatch (%).

Approximately 6-7 dB of immersion fidelity improvement have been enabled across both 360° sequences and all network bandwidth values considered in Figure 14. These advances can in turn enable much higher operational efficiency for a streaming system for 360° video delivery that integrates our methodology from Section IV, as our subsequent results demonstrate.

End-to-end system performance. We first examine the energy efficiency of our system framework, when streaming the 360° content to the mobile VR users. In Figure 15, we study the energy consumption of the several methods under comparison, as a function of the available storage capacity of the edge servers collocated with the small base stations. In this evaluation, we set the computing capability (power) $\bar{Z}_i$ of the edge server at each small base station i to 65W, following commonly adopted values for present microprocessors [51].

We can see that our approach considerably outperforms the two reference methods, demonstrating three to six times lower energy consumption across the entire range of x-axis values examined in Figure 15. These gains are enabled by our optimization framework that can optimally pool and allocate the transmission, storage, and computing resources available at the small base stations. Moreover, our approach enables much higher energy consumption versus storage capacity efficiency, for low values of the latter, as observed in Figure 15.

As expected, the reference method *Blasco2014* outperforms the other reference method *LRU* due to the optimization it implements, though its gains appear to be marginal. Finally, we also evaluated the performance of our framework when operating on network coding packets. We can see from Figure 15 that this introduces further energy consumption savings, as in this case the optimization (3) - (6) can be solved exactly, as explained earlier. This then leads to even higher utilization of the available system resources, as expected.

Next, in Figure 16, we study the energy consumption of the several methods under comparison, as a function of the available computing capacity of the edge servers collocated with the small base stations. In this evaluation, we set the storage capacity Z_i of the edge server at each small base station i to 100 GB. We can see again that our approach considerably outperforms the two reference methods, demonstrating three to seven times lower energy consumption across the entire range of x-axis values examined in Figure 16. These gains are enabled by our optimization framework that can optimally pool and allocate the transmission, storage, and computing resources available at the small base stations, as explained earlier. Moreover, our approach enables much higher energy consumption versus computing capacity efficiency, for low values of the latter, as observed in Figure 16.

Evaluating the performance of our framework on network coding packets leads to further energy consumption savings, as seen from Figure 16. This is an expected outcome equivalent to that observed in the context of the evaluation and results reported in Figure 15 earlier. These further benefits arise from the possibility to solve the optimization (3) - (6) exactly, thereby enabling even higher system resources' utilization efficiency, as explained before. The similar performance outcomes observed in Figure 15 and Figure 16 demonstrate the consistent benefits introduced by our framework against diverse key system parameters.

Finally, we evaluated the delivered immersion fidelity enabled by our approach across the different 360° videos streamed to the mobile VR users in the setting under consideration. These results are shown in Figure 17, which includes the corresponding performance results for the reference method *Blasco2014*. The equivalent findings for the second reference method *LRU* are not included in the figure, as they are lower than those for *Blasco2014*. We can see that across all five 360° videos considered in Figure 17 our approach enables consistently significant gains in immersion fidelity delivered to the mobile users, ranging between six to ten decibels of the viewport Y-PSNR for a mobile VR user.

Moreover, our approach enabled delivering some 360° videos also at higher spatial resolutions or temporal frame rates, thereby augmenting further the user's immersion quality of experience. In particular, the video Paris was delivered at temporal frame rate of 60 fps by our approach, as indicated in Figure 17, while the reference method *Blasco2014* could only stream it at 30 fps. Similarly, our approach enabled delivering the videos Basketball and Runner at higher 8K spatial resolution for the 360° panorama, as indicated in Figure 17, while *Blasco2014* could only stream them at 4K spatial resolution.

Packet transmission scheduling utility. We carry out two experiments to evaluate solely the efficiency of the 360° packet scheduling, proposed in Section VI to address prospective network transients at the small base stations, as described therein. In the first evaluation, we measure the reduction in

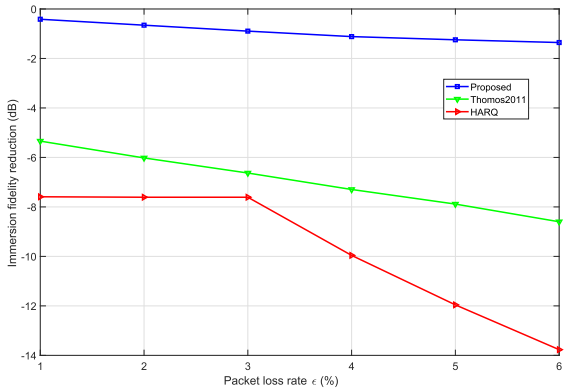

Fig. 19. Immersion fidelity reduction vs. packet loss rate ϵ (%).

delivered immersion fidelity when the compressed video packets of a 360° video to be transmitted are buffered at a small base station, due to an insufficient transmission capacity, and their outgoing transmissions need to be carefully scheduled. We consider that the packet loss rate ϵ on the respective downlink to the mobile VR user is zero in this evaluation.

In Figure 18, we show these results versus the respective mismatch between the data rate required by the 360° content and the actual available transmission capacity, expressed in percent of the first quantity. First, we can see that for the same set of buffered 360° packets, all three methods show no reduction in delivered immersion fidelity, in the case of no mismatch between the 360° content's required data rate and the available transmission capacity, as expected. However, once the network data rate mismatch starts increasing, we can see from Figure 18 that the performance of the reference methods *HARQ* and *Thomos2011* considerably decreases. On the other hand, our approach enables robust and consistent performance across the entire range of mismatch values evaluated, due to its rate-distortion optimized design that integrates the importance of every 360° packets for the immersion fidelity delivered to the user, as formulated in Section VI.

Concretely, we observe a drop of close to seven decibels in immersion fidelity for a mismatch of 6%, in the case of *HARQ*. This outcome is expected as *HARQ* treats every packet equally, when deciding on their transmissions, and thus it is oblivious to the impact that omitting the transmission of a packet may have on the resulting immersion fidelity experienced by the respective mobile VR user. The second reference method *Thomos2011* offers a more graceful degradation in performance, as the data rate mismatch increases, as evident from Figure 18, due to the partial robustness it introduces. However, the immersion fidelity reduction it demonstrates is still considerable and reaches up to 4.2 decibels. On the other hand, an immersion fidelity reduction of less than 0.5 decibels is maintained by our approach throughout the entire range of mismatch values examined in Figure 18.

In the second evaluation, we consider that there is no mismatch between the required data rate for the buffered packets and the available transmission capacity, however, the respective downlink to the mobile VR user features non-zero packet loss rate ϵ. Here, the three methods under comparison need to decide the transmission scheduling of the buffered 360° packets such that it includes prospective retransmissions of those packets lost during their original transmission. We measure the resulting reduction in delivered immersion fidelity for each method against the packet loss rate ϵ. These results are shown in Figure 19.

We can see that our approach again enables robust and consistent performance across the entire range of loss rates examined in the figure. This is due to its rate-distortion optimized design that rigorously integrates the importance of every 360° packets for the immersion fidelity delivered to the user and the impact of prospective packet loss, when scheduling the transmissions of outgoing packets. Thus, our approach maintains an immersion fidelity reduction that does not exceed 1.5 decibels, even at 6% downlink packet loss.

On the other hand, the performance of the reference methods *HARQ* and *Thomos2011* considerably decreases, as the packet loss rate increases, as seen from Figure 19. In particular, we observe an immersion fidelity reduction of close to 14 dB and 8.2 dB respectively for *HARQ* and *Thomos2011*, for packet loss rate of six percent. The considerably degraded performance of *HARQ* is due to the same reasons explained earlier, and it features the well-known cliff effect of traditional error protection (FEC) [52], as witnessed by Figure 19. The error-adaptive design of *Thomos2011* helps it provide more graceful performance degradation relative to *HARQ*. However, since its operation is data-agnostic, its performance is still notably penalized over our approach.

VIII. Conclusion

We have explored a novel communications system that integrates for the first time scalable multi-layer 360° video tiling, viewport-adaptive rate-distortion optimal resource allocation, and VR-centric edge computing and caching, to enable next generation high-quality untethered VR streaming. Our system comprises a collection of 5G small cells that can pool their communication, computing, and storage resources to collectively deliver scalable 360° video content to mobile VR clients at much higher quality. The major contributions of the paper are the rigorous design of multi-layer 360° tiling and related models of statistical user navigation, analysis and optimization of edge-based multi-user VR streaming that integrates viewport adaptation and server cooperation, and base station 360° video packet scheduling. We also investigated the possibility of network coded data operation and its implications for the analysis, optimization, and system performance we pursue in this setting. The advances introduced by our framework over the state-of-the-art comprise considerable gains in delivered immersion fidelity, featuring much higher 360° viewport peak signal to noise ratio (PSNR) and VR video frame rates and spatial resolutions. Integrating our advances in practice represents prospective fruitful future work.

References

[1] Digi-Capital Research. (Jul. 2016). *Virtual, Augmented and Mixed Reality are the 4th Wave*. [Online]. Available: https://www.digi-capital.com
[2] J. G. Apostolopoulos, P. A. Chou, B. Culbertson, T. Kalker, M. D. Trott, and S. Wee, "The road to immersive communication," *Proc. IEEE*, vol. 100, no. 4, pp. 974–990, Apr. 2012.
[3] *Worldwide Revenues for Augmented and Virtual Reality Forecast to Reach $162 Billion in 2020*. Bus. Wire, San Francisco, CA, USA, Aug. 2016.
[4] B. Begole, "Why the Internet pipes will burst when virtual reality takes off," *Forbes Mag.*, Feb. 2016.

[5] E. Knightly, "Scaling Wi-Fi for next generation transformative applications," Keynote Presentation, *IEEE INFOCOM*, May 2017.
[6] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.
[7] 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 HotMobile Workshop*, Tempe, AZ, USA, Feb. 2018, pp. 7–12.
[8] J. Chakareski, "Uplink scheduling of visual sensors: When view popularity matters," *IEEE Trans. Commun.*, vol. 63, no. 2, pp. 510–519, Feb. 2015.
[9] R. Vasudevan, Z. Zhou, G. Kurillo, E. Lobaton, R. Bajcsy, and K. Nahrstedt, "Real-time stereo-vision system for 3D teleimmersive collaboration," in *Proc. IEEE ICME*, Jul. 2010, pp. 1208–1213.
[10] M. Hosseini and G. Kurillo, "Coordinated bandwidth adaptations for distributed 3D tele-immersive systems," in *Proc. 7th ACM Int. Workshop Massively Multiuser Virtual Environ.*, 2015, pp. 13–18.
[11] G. Cheung, A. Ortega, and N.-M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 744–761, Mar. 2011.
[12] J. Chakareski, V. Velisavljevic, and V. Stankovic, "User-action-driven view and rate scalable multiview video coding," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3473–3484, Sep. 2013.
[13] J. Chakareski, "Wireless streaming of interactive multi-view video via network compression and path diversity," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1350–1357, Apr. 2014.
[14] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–7.
[15] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Optimal set of 360-degree videos for viewport-adaptive streaming," in *Proc. ACM Int. Conf. Multimedia*, Oct. 2017, pp. 934–951.
[16] 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.*, Beijing, China, Sep. 2017, pp. 2174–2178.
[17] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "Improving virtual reality streaming using HTTP/2," in *Proc. 8th ACM Multimedia Syst. Conf.*, Jun. 2017, pp. 225–228.
[18] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2016, pp. 107–110.
[19] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[20] P. Blasco and D. Gunduz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE ICC*, Jun. 2014, pp. 1897–1903.
[21] S. Martello and P. Toth, *Knapsack Problems*. New York, NY, USA: Wiley, 1990.
[22] K. Shanmugam, N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.
[23] E. Bastug, M. Bennis, and M. Debbah, "Cache-enabled small cell networks: Modeling and tradeoffs," in *Proc. IEEE ISWCS*, Aug. 2014, pp. 649–653.
[24] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen, and O. Spatscheck, "To cache or not to cache: The 3G case," *IEEE Internet Comput.*, vol. 15, no. 2, pp. 27–34, Mar. 2011.
[25] H. Ahlehagh and S. Dey, "Video caching in radio access network: Impact on delay and capacity," in *Proc. IEEE WCNC*, Apr. 2012, pp. 2276–2281.
[26] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. Diggavi, "Hierarchical coded caching," in *Proc. IEEE ISIT*, Jul. 2014, pp. 2142–2146.
[27] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 1077–1081.
[28] A. Khreishah, J. Chakareski, and A. Gharaibeh, "Joint caching, routing, and channel assignment for collaborative small-cell cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2275–2284, Aug. 2016.
[29] A. Khreishah and J. Chakareski, "Collaborative caching for multicell-coordinated systems," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 257–262.
[30] J. Chakareski, "VR/AR immersive communication: Caching, edge computing, and transmission trade-offs," in *Proc. ACM SIGCOMM Workshop Virtual Reality and Augmented Reality Netw.*, Aug. 2017, pp. 36–41.
[31] 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 Syst.*, Jun. 2017, pp. 261–271.
[32] R. Monnier, R. van Brandenburg, and R. Koenen, "Streaming UHD-quality VR at realistic bitrates: Mission impossible?" TiledMedia, Rotterdam, The Netherlands, White Paper, May 2017.
[33] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "OpTile: Toward optimal tiling in 360-degree video streaming," in *Proc. ACM Multimedia*, Oct. 2017.
[34] Y. Sun, A. Lu, and L. Yu, "Weighted-to-spherically-uniform quality evaluation for omnidirectional video," *IEEE Signal Process. Lett.*, vol. 24, no. 9, pp. 1408–1412, Sep. 2017.
[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 Syst. Video Technol.*, vol. 26, no. 1, pp. 20–34, Jan. 2016.
[36] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 56–67, Oct. 1998.
[37] N. Thomos, J. Chakareski, and P. Frossard, "Prioritized distributed video delivery with randomized network coding," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 776–787, Aug. 2011.
[38] V. Vazirani, *Approximation Algorithms*, 2nd ed. New York, NY, USA: Springer-Verlag, 2003.
[39] R. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1962.
[40] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Programming*, 3rd ed. Nashua, NH, USA: Athena Scientific, Feb. 1997.
[41] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, no. 3, pp. 399–417, Jun. 1963.
[42] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. INFOCOM*, Mar. 1999, pp. 126–134.
[43] X. Liu, Y. Huang, L. Song, R. Xie, and X. Yang, "The SJTU UHD 360-degree immersive video sequence dataset," in *Proc. IEEE Int. Conf. Virtual Reality Vis.*, Oct. 2017, pp. 400–401.
[44] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, "Viewport-driven rate-distortion optimized 360° video streaming," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. .
[45] X. Corbillon, F. De Simone, and G. Simon, "360-degree video head movement dataset," in *Proc. ACM Multimedia Syst. Conf.*, Jun. 2017, pp. 1–7.
[46] A. Fei, G. Pei, R. Liu, and L. Zhang, "Measurements on delay and hop-count of the Internet," in *Proc. Globecom Internet Mini-Conf.*, 1998.
[47] J. Huang, F. Qian, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in *Proc. ACM MobiSys*, 2012, pp. 225–238.
[48] V. Sivaraman, A. Vishwanath, Z. Zhao, and C. Russell, "Profiling per-packet and per-byte energy consumption in the NetFPGA gigabit router," in *Proc. IEEE INFOCOM Workshops*, Apr. 2011, pp. 331–336.
[49] R. Comroe and D. Costello, "ARQ schemes for data transmission in mobile radio systems," *IEEE J. Sel. Areas Commun.*, vol. 2, no. 4, pp. 472–481, Jul. 1984.
[50] K. Piamrat, C. Viho, J.-M. Bonnin, and A. Ksentini, "Quality of experience measurements for video streaming over wireless networks," in *Proc. IEEE Int. Conf. Inf. Technol., New Gener.*, Apr. 2009, pp. 1184–1189.
[51] *CPU Power Dissipation*. [Online]. Available: https://en.wikipedia.org/wiki/CPU_power_dissipation/
[52] S. Rane, P. Baccichet, and B. Girod, "Systematic lossy error protection of video signals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 10, pp. 1347–1360, Oct. 2008.

Jacob Chakareski (Senior Member, IEEE) completed his Ph.D. degree in electrical and computer engineering at Rice University and Stanford University. He held research appointments at Microsoft, HP Labs, and EPFL, and served on the Advisory Board of Frame, Inc. He is an Associate Professor with the Ying Wu College of Computing, NJIT, where he leads the Laboratory for VR/AR Immersive Communication (LION). His research has been supported by the NSF, NIH, AFOSR, Adobe, Tencent Research, NVIDIA, and Microsoft. His research interests span networked virtual and augmented reality, UAV-IoT sensing and networking, real-time reinforcement learning, 5G wireless edge computing and caching, millimeter wave and free-space optical wireless technologies, ubiquitous immersive communication, and societal applications. He is the organizer of the first NSF Visioning Workshop on networked VR/AR communications. He received the Swiss NSF Career Award Ambizione in 2009, the Best/Fast-Track Paper Award from IEEE GLOBECOM 2016 and IEEE ICC 2017/2018, the AFOSR Faculty Fellowship in 2016 and 2017, and the Adobe Data Science Faculty Research Award in 2017 and 2018.