Cooperative Tile-based 360-degree Panoramic Streaming in Heterogeneous Networks using Scalable Video Coding

Xiaoyi Zhang, Xinjue Hu, Ling Zhong, Shervin Shirmohammadi, Lin Zhang

***Abstract*—The use of high quality 360-degree panoramic video is booming in the video industry. However, existing schemes for smartphones suffer from significant bandwidth consumption as they transmit entire panoramic views in very high resolutions. This demand for bandwidth becomes even more problematic when multiple adjacent smartphones compete to access the same content, which further challenges a wireless network's capacity, on which the available bandwidth fluctuates much more than wired networks. In this paper, we propose a cooperative streaming scheme for tile-based 360-degree video using Scalable Video Coding (SVC) to maximize a group of users' Quality of Experience (QoE). We formulate an optimization problem to choose optimal downloading and sharing subsets from a set of all requested SVC layers of tiles to maximize the effective quality of the users' viewport while meeting the feasibility of the bandwidth of heterogeneous networks. We then show that the problem is NP-hard and compose a heuristic approach. In the approach, we rank the SVC layers based on the aggregated group-level preference to guide the devices' downloading and sharing activities. A prototype on the Android platform is developed to test the approach's performance, and the real-world results show that our proposed scheme outperforms baseline alternatives.**

***Index Terms*—360-degree Panoramic Video, Cooperation, Heterogeneous Networks.**

I. Introduction

As immersive video technologies have advanced over the past few years, 360-degree panoramic video has also increased in popularity. Because it is a centerpiece of the virtual reality (VR) industry, 360-degree video is predicted to garner a huge market of $120 billion by 2020 [1]. Users may enjoy 360-degree panoramic videos on head-mounted displays (HMD), such as the Google Cardboard [2] or Samsung Gear VR [3], anytime and anywhere.

These videos are recorded by omnidirectional cameras to support an immersive experience. When a device is rendering a 360-degree video with equirectangular projection [4], the video frame's pixels are mapped into a spherical surface, and the viewer at the center of the sphere enjoys an immersive experience. The field of view (FOV) of available HMDs ranges between 96° and 110° [5], so the device only needs to display part of the video frame. However, current commercial schemes deployed by flagship video platforms (e.g., YouTube [6] and Facebook [7]) treat 360-degree panoramic videos as ordinary videos. Since the device only displays a portion of the video, transmitting the whole video content puts an unnecessary burden on a network's bandwidth.

Recent studies have proposed tile-based schemes [8]–[14] to reduce the required bandwidth for a single consumer. They spatially crop the 360-degree video into *tiles* and independently encode each tile into multiple versions at various bitrates. HMDs can adaptively select the tiles' quality level depending on the coverage of the current FOV. The video area being watched is called the *viewport*, and the tiles in the viewport should always be delivered in high quality.

Although tile-based schemes help save bandwidth, their efforts are not sufficient to ensure the quality of the video when multiple adjacent users are simultaneously streaming a 360-degree video via a cellular network. This could be a group of users in a classroom, in an office space, at home, or in a moving train. In this type of scenario [15]–[17], these schemes induce competition for bandwidth, so a group of non-cooperating devices will likely suffer from network congestion. To alleviate this problem, researchers have proposed cooperative streaming, whereby a group of adjacent users, close enough to form a Mobile Ad hoc Network (MANET), watch the same video [16]–[21]. However, these cooperative schemes are designed for ordinary videos and would perform less impressively in a tile-based scheme, as the consumers of 360-degree panoramic videos demand different tiles depending on the direction of their viewports. A naive application of the sharing scheme, which would simply download the whole video file and share it with everyone in the group, would result in redundancy. Therefore, the key problem in increasing efficiency and Quality of Experience (QoE) for the group is identifying potential mutual demands from the group to economically allocate the network resources, and this key problem remains unresolved.

In this paper, we target a scenario where a group of users, who are physically close enough together to form a MANET, are watching the same 360-degree video from different angles. To support this scenario, we propose a cooperative streaming scheme to optimize the QoE of the group. In our scheme, the server encodes each tile-based video using the quality adaptation method of the Scalable Video Coding (SVC) standard, which splits each ordinary tile into a base layer and multiple base-dependent enhancement layers to enable decoding of the tile despite the potential unavailability (due to network loss) of some of the high-level layers. Then, the server chooses a

X. Zhang, X. Hu, L. Zhang are affiliated with Beijing University of Posts and Telecommunications, China.L. Zhong is affiliated with Yale University, USA.S. Shirmohammadi is affiliated with University of Ottawa, Canada.

pair of downloading and sharing subsets from the set of all SVC layers of the tiles. The mobile devices used in the group download these layers separately via their cellular network and share them through the MANET.

To take advantage of the network context, we formulate the search for the optimal scheme as a maximization problem of the video quality of the users' displayed viewports, constrained by the feasibility of the bandwidth of heterogeneous networks. To solve this NP-hard problem, we propose a heuristic approach to rank the SVC layers by assessing their quality, layer level, probability of being watched, and the potential of being demanded by multiple users. We then develop a prototype on the Android platform to test the performance of our scheme and show that it outperforms baseline alternatives in the real world.

The main contributions of this article can be summarized as follows:

- We propose, for the first time, a cooperative video streaming scheme specifically for 360-degree panoramic video.
- We model and formulate a group level optimization problem for our scheme and propose to rank the SVC layers to find a heuristic solution.
- We design a ranking mechanism that takes individual requests from the consumers and optimally allocates resources to provide high-quality video.
- We provide proof of concept through prototype implementation and experiments on several Android smartphones, demonstrating the efficiency of our scheme.

The remainder of this paper is organized as follows. Section II summarizes related research from the literature. Section III describes the architecture of the proposed scheme and formulates the optimization problem. Section IV demonstrates the proposed cooperative approach that ranks the SVC layers, while Section V presents the real-world results and compares the performance of our approach with other existing approaches.

II. Related Works

The key components of our work include tile-based 360-degree panoramic video streaming and cooperative video transmission for multiple users. We summarize and compare our work with previous works that share these components in this section.

A. Transmitting Tile-based 360-degree Panoramic Video

Tile-based streaming is a suitable approach for 360-degree panoramic video as the visible area of the video to a single user is a strict and relatively small subset of the entire video file. Previous studies have constructed efficient systems to transmit tile-based 360-degree panoramic video [8]–[13]. In particular, Zare et al. [8] proposed a system to stream panoramic video that stores two versions of the same video content at different resolutions. Each version of the video was divided into multiple tiles using the High Efficiency Video Coding (HEVC) standard. The proposed system transmitted a set of tiles in the highest captured resolution and the remaining parts in low-resolution. Qian et al. [9] also proposed a system that delivers tile-based 360-degree panoramic videos. It considered the movement of the viewer's head and predicted the viewport using linear regression (LR). This system only disseminates video tiles that are predicted to be visible to the user to reduce bandwidth consumption. Directly applying these systems for multiple users watching the same 360-degree video causes some practical issues: (1) It fetches the best results based on their ability, and will lead to competition among adjacent devices; (2) It ignores the potential of sharing tiles within a group, which may be a nonempty subset of all tiles even when the users are watching different parts of the same video. Little work has been done on solving this optimization problem from a group of users' perspective. Ahmadi et al. [13], as the only existing multicast scheme in this topic, compromised individual-level quality for balanced video quality across users. Our work fills this gap in the 360-degree video streaming context by bringing in the group-level cooperation, which costs more bandwidth but maximizes video quality.

B. Cooperative Video Transmission for Multiple Users

Cooperation is a natural approach to reach efficiency in multiple user scenarios, in which the parts of the video demanded by more than one user may be downloaded only once and shared. Cooperative systems take advantage of the existence of intersections in the users' demands to make a better arrangement for all users than solving individual-level optimization problems. Existing studies, focusing on cooperation when watching ordinary videos, have proposed a few schemes to make use of this type of sharing [16]–[21]. Le et al. [16] introduced a cooperative system in which mobile devices cooperate to efficiently utilize all network resources and adapt to varying wireless-network conditions. The problem was formulated using a Network Utility Maximization (NUM) framework. In the local WiFi network, overhearing combined with network coding was exploited to improve the user's experience. Our previous work [19] also proposed a device-to-device video streaming system via hybrid cellular and ad hoc networks. The system balanced the gradually diminishing marginal group-level QoE and linear energy consumption by assigning downloading tasks to some of the mobile devices in the group and letting them share the downloaded video. Downloading the entire 360-degree panoramic video file wastes a substantial amount of network resources, considering the fraction of the file that will actually be watched. While the common demand in the case of ordinary videos is essentially the entire video file and is trivial to identify, the exact set of the shared Regions of Interest (ROIs) of a 360-degree panoramic video are difficult to identify, predict, and accommodate. Our work builds upon our previous framework [17], [19] for ordinary videos to incorporate the features of 360-degree panoramic video to (1) identify potential shared ROI in the users' predicted viewports and (2) prioritize the downloading order of different parts of the video to account for the limited bandwidth for transmitting 360-degree panoramic videos.

III. Cooperative Panoramic Video Architecture

We begin this section with an overview of the framework architecture and optimization problem in the system. The system

consists of a group of users indexed by $i = 1, 2, 3, \cdots, I$. The users, who are within close physical proximity, are watching the same 360-degree panoramic video. The devices they use to watch the video form heterogeneous networks that include the cellular network and the local MANET. The video is processed into tiles and generated following the SVC standard. The video layers of the same tile encoded by SVC provide more variation in downloading and sharing plans, and this flexibility is good for fluctuating bandwidth conditions and innovative ways of sharing common demands among devices. The system, taking advantage of the overlapping nature of the tiles requested by different devices, assigns downloading and sharing tasks to reduce the collective consumption of the bandwidth. Specifications of the system should be the solution of an optimization problem that allocates tasks to minimize the average distortions of the visible area of the video, constrained by bandwidth capacities. The list of symbols and notations used in this paper is provided in Table I.

TABLE I
NOTATION TABLE

Symbol	Definition
i	User's index, $i = 1, 2, 3, \cdots, I$
S_j	Segment j of a video file, $j = 1, 2, \cdots, J$
τ	Time duration of a video segment
T_{jwh}	Tile wh of segment j, $w = 1, 2, \cdots, W$ and $h = 1, 2, \cdots, H$
L_{jwhk}	SVC layer k of T_{jwh}, $k = 1, 2, \cdots, K$
$\delta(l)$	Data amount of layer l
$q(l)$	Cumulative video quality of the tile up to layer l
$\Delta q(l)$	Marginal quality of layer l
ξ^i_{jwhk}	Allocation of downloading task of layer L_{jwhk} on user i
ξ^{wifi}_{jwhk}	Allocation of sharing task of layer L_{jwhk}
$Occ^i(l)$	Indicator for user i's occupancy of layer l
$D^i(l)$	Amount of marginal video distortion on user i's device caused by layer l
$\psi^i_j := obs^i_j(\{t\})$	Set of observed tiles among all tiles in set $\{t\}$ in segment j of user i
$\hat{D}^i_j$	Video distortion of user i's viewport in segment j
ζ^i	Cellular bandwidth of user i
ζ^{wifi}	MANET bandwidth
$\Delta Occ^{i,1}(l)$	Partial derivative of $Occ^i(l)$ with respect to ξ^1_{jwhk}
$\Delta D^{i,1}(l)$	Partial derivative of $D^i(l)$ with respect to ξ^1_{jwhk}
$\Delta \hat{D}^{i,1}_{jwhk}$	Partial derivative of $\hat{D}^i_j$ with respect to ξ^1_{jwhk}
$\Delta\text{AVD}^1_{jwhk}$	Partial derivative of the average video distortion with respect to ξ^1_{jwhk}
$\Delta Occ^{i,\text{wifi}}(l)$	Partial derivative of $Occ^i(l)$ with respect to ξ^{wifi}_{jwhk}
$\Delta\text{AVD}^{\text{wifi}}_{jwhk}$	Partial derivative of the average video distortion with respect to ξ^{wifi}_{jwhk}
$\theta(P, Q)$	Euclidean distance between the points P and Q
$dist^i(l)$	Distance between the tile of layer l and the user i's viewport
V	Center point of the viewport
L_{center}	Center point of the layer
$bin(l)$	Index of the bin that layer l belongs to
κ	Fixed width of a bin
$rank^i(l)$	User i's device-level ranking score for layer l
η^i_j	Ratio between the total quality of all layers of all tiles in the current viewport of user i and the total marginal quality of the base layers of all tiles on the sphere, within segment j
$index^i(l)$	Device-level order index of user i for layer l
$rank_g(l)$	Group-level ranking score of layer l
$rank_s(l)$	Sharing ranking score of layer l
AVD_x	Average video distortion of scheme x
σ_j	Standard deviation of tile-level video distortion in segment j
Rate^i_j	Bitrate of visible tiles of video segment j to user i
QoE^J_1	Composite QoE of video segments 1 to J

A. Preparation for Tile-based SVC Video Sources

The essential difference between ordinary videos and 360-degree panoramic videos is that the consumers of the latter only watch portions of the video at any given time. However, since 360-degree panoramic video consumers move their viewports around, downloading the video areas that are not currently watched but has a potential to be watched in the near future is as important as downloading the current viewport. When taking these potential areas into account, the downloading tasks overlap across multiple consumers even if their current viewports do not. The dynamics in the position of the viewports and the potential overlapping demand for a video area motivate us to crop the video temporally and spatially.

Having said that, identifying the overlapping areas is inadequate because the qualities of the areas in demand vary across consumers. Almost all existing schemes storing non-scalable videos (e.g., those which follow the Advanced Video Coding (AVC) standard), fail to differentiate based on importance. Not being able to extract low-quality video from high quality video, these existing schemes have no choice but to transmit both low and high quality versions of the tile and cannot make the intersection of this situation because the two files are different.

In this subsection, we exploit the possibility of economically allocating resources through the development of a cooperative scheme. In the scheme, we cut the video file into three dimensions: we split the duration into time segments to adjust for the moving dynamics, crop the segment into tiles to nail down the overlapping area, and encode the tile into quality-adaptive SVC layers. This multi-dimensional fragmentation of a video file adds to service flexibility by allowing for heterogeneous users' demands for various video tiles and quality levels while enabling economical consumption of network resources.

The processing of the video source is described in Fig. 1. The raw file of the video is temporally segregated with equal time durations τ into segments, and we denote the segments by $\{S_j\}_{j=1}^J$. This temporal segregation helps keep up with users' constantly changing requirements as it provides opportunities for readaptation in each segment. Then, the video in each segment is spatially cropped into tiles and indexed by $\{T_{jwh}\}_{w=1,h=1}^{w=W,h=H}$. Each tile is encoded into several layers using quality adaptive SVC, and the set of layers of tile T_{jwh} is $\{L_{jwhk}\}_{k=1}^K$. We denote the data amount of layer L_{jwhk} by $\delta(L_{jwhk})$. As a direct result of SVC, the base layer, indexed by L_{jwh1}, can be decoded to play independently, while an enhancement layer requires the layers of the same tile with lower indices.

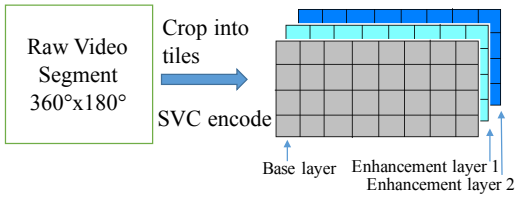


Fig. 1. Tiling and Encoding of a Panoramic Video Segment

When the downloading process is limited by network resources, and there is a failure in downloading a tile at layer $k+1$, this tile can still be smoothly displayed at layer k. We use two objective quality measurements, the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM), to evaluate video quality, and the quality of the video produced by layers 1 to k is $q(L_{jwhk})$. In particular, we let $q(L_{jwh0}) = 0$. To quantify the increase in the quality generated by each additional layer, we define the marginal quality that layer k yields as follows:

$$\Delta q(L_{jwhk}) = q(L_{jwhk}) - q(L_{jwh,k-1}) \tag{1}$$

B. Cooperative Video Streaming System

The system organizes the devices and SVC layers of the video as follows. Each device connects to the server independently via a cellular network, and the devices communicate with each other through the local MANET in a broadcast manner as described in our previous works [17], [19]. An outline of the architecture is shown in Fig. 2. The cellular network allows each device to download from the external server, and the MANET enables them to share the data locally. The server receives separate requests from the devices and observes their potential to cooperate. By considering the bandwidth limitations of both the cellular network and the MANET, as well as the monetary cost of the devices' data plans, the server proposes an allocation plan for this group.

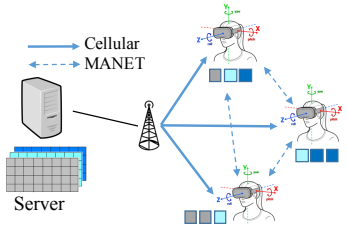


Fig. 2. Architecture of the Cooperative SVC Streaming System

The allocation of downloading tasks among devices in a group can be represented by a set of indicators $\{\xi^i_{jwhk}\}$ in which i is a user, j is a video segment, wh is a tile, and k is a SVC layer. The indicator ξ^i_{jwhk} is 1 if layer L_{jwhk} is downloaded by device i. In parallel, the allocation of sharing tasks can be represented by another set of indicators $\{\xi^{\text{wifi}}_{jwhk}\}$. The indicator ξ^{wifi}_{jwhk} is 1 if L_{jwhk} is shared within the group. A layer can be shared only if it is downloaded by at least one of the users.

$$0 \le \xi^{\text{wifi}}_{jwhk} \le \sum_{i=1}^{I} \xi^i_{jwhk} \tag{2}$$

Given such an allocation plan, the server and connected devices process this plan in a cyclical manner. As illustrated in Fig. 3, when the devices decode and play video segment j, they share segment $j+1$ on the local MANET while downloading segment $j+2$ and communicate with the server to finalize the allocation plan of segment $j+3$. It should be mentioned that our scheme does not suffer from video jitter caused by waiting, nor does it use a playback buffer to eliminate video starvation. A visible tile without any downloaded video layer will be rendered as a black tile. This setting is specific to our choice on SVC encoding and a time-block structure with strict deadlines. We leave the task of minimizing the effect of this restriction on video quality to our proposed optimization problem.

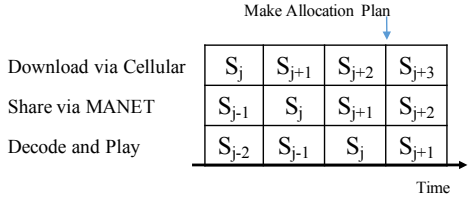


Fig. 3. Sequence of Time Block in the Cooperative System

C. Optimization Problem

Next, we formulate the optimization problem in the environment we have set up in subsections III-A to III-B. The problem describes the objective and the constraints faced by the server, and it takes the following conceptual form:

It chooses an allocation plan to minimize

- Average Video Distortion (AVD);
- downloading task duplication,

subject to constraints, including

- the amount of data downloaded by each device does not exceed its cellular network capacity;
- the total amount of shared data does not exceed the capacity of the local MANET.

In the context of this paper, we specify the components of the above optimization problem as follows.

1) The Expression for AVD: We define the video distortion of a particular device as the gap between the acquired video quality on this device and the highest quality of the video stored on the server. AVD is the unweighted average of the video distortion of each device in the group. Mathematically, the video distortion of a device is a function of the set of marginal qualities and the set of accessibility indicators of the SVC layers. The marginal quality of a layer is defined in Eq. 1, and the overall occupancy of a certain layer may be represented by the following binary variable:

$$\begin{aligned} &Occ^i(L_{jwhk}) \\ =&1 - (1 - \xi^i_{jwhk}) \cdot \{1 - \xi^{\text{wifi}}_{jwhk} \cdot [1 - \prod_{i' \neq i} (1 - \xi^{i'}_{jwhk})]\} \\ =&\xi^i_{jwhk} + \xi^{\text{wifi}}_{jwhk} \cdot [1 - \xi^i_{jwhk} - \prod_{i'=1}^{I} (1 - \xi^{i'}_{jwhk})] \end{aligned} \tag{3}$$

in which the first summand is the indicator for whether device i is the downloader of this layer, and the second summand is the indicator for whether device i can successfully acquire this layer from the local MANET. The event of successfully acquiring this layer from the local MANET happens only when another device in the group downloads this layer, and the layer is shared on the local MANET.

We obtain the expression for video distortion as follows:

$$D^i(L_{jwhk}) = \Delta q(L_{jwhk}) \cdot [1 - \prod_{l=1}^{k} Occ^i(L_{jwhl})] \quad (4)$$

The product of the individual occupancy indicators in Eq. 4, where all layers of the same tile with lower quality levels are downloaded, is derived from the dependency of the SVC decoding procedure. An SVC layer is actually accessible to the device only if the device also has all of its underlying layers. Therefore, the distortion is positive when any of the underlying layers of this tile are missing. Since user i only observes part of the 360-degree panoramic video, the set of tiles affecting its QoE is a strict subset of all tiles, and only tiles in this subset should contribute to the calculation of the AVD. We define this effective subset as follows:

$$\psi_j^i = obs_j^i(\{T_{jwh}\}) \quad (5)$$

in which $obs_j^i(\{T_{jwh}\})$ is a subset of $\{T_{jwh}\}$, and every tile in this subset is observed by user i.

The video distortion of the viewport of user i throughout segment j is defined as follows:

$$\hat{D}_j^i = \frac{1}{|\psi_j^i|} \sum_{T_{jwh} \in \psi_j^i} \sum_{k=1}^{K} D^i(L_{jwhk}) \quad (6)$$

in which $|\psi_j^i|$ is the cardinality of the set ψ_j^i.

2) Task Duplication Constraint: In this paper, we simplify this second order objective into a constraint, such that each SVC layer is downloaded no more than once by the group as a whole (Eq. 7). We obtain the first constraint as in the problem statement (Eq. 10).

$$\sum_{i=1}^{I} \xi_{jwhk}^i \leq 1 \quad (7)$$

3) Capacity constraints: The estimated cellular bandwidth of device i and the group's estimated MANET bandwidth are denoted as ζ^i and ζ^{wifi}, respectively. With the notation defined above, we may express the two constraints as follows:

- The downloaded data amount on the device of user i does not exceed the bandwidth capacity for this user.

$$\sum_{\forall w,h,k} \delta(L_{jwhk}) \cdot \xi_{jwhk}^i < \tau \cdot \zeta^i \quad for\ all\ i \quad (8)$$

- The total amount of shared data does not exceed the capacity of the local MANET.

$$\sum_{\forall w,h,k} \delta(L_{jwhk}) \cdot \xi_{jwhk}^{\text{wifi}} < \tau \cdot \zeta^{\text{wifi}} \quad (9)$$

With the notation defined above, we may express the two capacity constraints as the second and third constraints in the problem statement (Eq. 10).

4) Problem Statement: In summary, the optimization problem that we vaguely defined at the beginning of this subsection is now depicted by the following constrained minimization problem.

$$\begin{aligned} \min_{\{\xi_{jwhk}^i\},\{\xi_{jwhk}^{\text{wifi}}\}} \ & \frac{1}{I} \sum_{i=1}^{I} \hat{D}_j^i \\ s.t. \ \ & 0 \leq \xi_{jwhk}^{\text{wifi}} \leq \sum_{i=1}^{I} \xi_{jwhk}^i \leq 1 \qquad for\ \forall w,h,k \\ & \sum_{\forall w,h,k} \delta(L_{jwhk}) \cdot \xi_{jwhk}^i < \tau \cdot \zeta^i \quad for\ all\ i \\ & \sum_{\forall w,h,k} \delta(L_{jwhk}) \cdot \xi_{jwhk}^{\text{wifi}} < \tau \cdot \zeta^{\text{wifi}} \end{aligned} \quad (10)$$

Remark. *As described in Subsection III-C, this problem for segment j is formulated and solved when segment $j-3$ is played, so the outcome of its solution is realized 3τ after it is solved.*

The mathematical expression of the optimization problem allows us to discuss its computational complexity from the properties of this type of math problem. The deterministic version of this problem is a binary integer linear programming (BILP) problem and, thus, is NP-hard. As noted in the remark, the uncertainty introduced by the unknown future ψ_j^i makes the problem a linear combination of a set of BILP problems, one for each possibility with probability density being its coefficient. A linear combination of a list of BILP problems is also BILP, so it may be solved through a dynamic programming approach. However, solving one BILP problem using dynamic programming takes a long time to converge as the run time of its worst-case scenario is exponential, let alone a huge set of problems. Given the instant nature of this type of 360-degree panoramic video, a prompt solution is very much desired and a long convergence process is not affordable. The heuristic algorithm we proposed in this paper to quickly solve the optimization problem obtains a close-to-optimal result with a practical convergence rate for efficient online operation.

IV. Cooperative Ranking Approach

Considering that the BILP problem described in the previous section is NP-hard, we cannot obtain an analytical solution for the optimization problem using basic algebraic operations in a timely manner. Fortunately, the desirable solution can be approximated by some heuristic approaches, and the time consumption of the approaches suits the context of this paper. The architecture of our proposed cooperative ranking approach is presented in Fig. 4.

A. From the Optimization Problem to the Ranking Approach

To draw a direct connection between the mathematical definition of the optimization problem and the features of the heuristic approach proposed in this section, we process the constrained minimization problem as stated in Eq. 10 to equalize it with the rationale behind our approach. The optimization problem consists of an objective function with two capacity constraints, one for downloading and one for sharing. The two parts of the task assignment can be considered separately. For each part, finding the optimal set of tasks with respect to its constraint is equivalent to providing assignments to elements

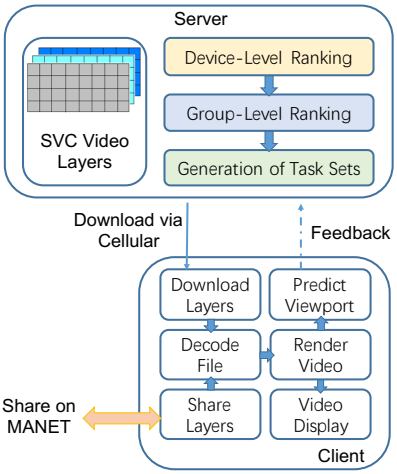


Fig. 4. Cooperative SVC Streaming System with Ranking Approach

in the global set of tasks individually until the constraint binds or a cost-benefit inequality is violated. Therefore the optimization problem can be thought of as a problem that ranks all elements in the set of tasks by their priority. To transform the mathematical formulation of the optimization problem into a heuristic approach, it suffices to find the rules to determine the ranking order between any pair of tasks. To characterize the rules from the optimization problem, the partial derivative of the objective function with respect to a specific assignment indicator is derived to reveal the marginal effect of the assignment to the optimization process.

Without loss of generality, we consider the marginal contribution of a downloading assignment ξ^1_{jwhk} to the objective function. Let us denote the marginal effect of ξ^1_{jwhk} on the occupancy indicator $Occ^i(L_{jwhk})$ by $\Delta Occ^{i,1}(L_{jwhk})$. Then, by Eq. 3, we have its expression as follows:

$$\begin{aligned}&\Delta Occ^{i,1}(L_{jwhk})\\ =&\frac{\partial}{\partial \xi^1_{jwhk}} Occ^i(L_{jwhk})\\ =&\begin{cases}1-\xi^{\text{wifi}}_{jwhk}+\xi^{\text{wifi}}_{jwhk}\cdot\prod\limits_{i'\neq 1}(1-\xi^{i'}_{jwhk}) & \text{if } i=1\\ \xi^{\text{wifi}}_{jwhk}\cdot\prod\limits_{i'\neq 1}(1-\xi^{i'}_{jwhk}) & \text{if } i\neq 1\end{cases}\end{aligned} \tag{11}$$

Next, we denote the marginal effect of ξ^1_{jwhk} on the video distortion of a layer $D^i(L_{jwhk})$ by $\Delta D^{i,1}(L_{jwhk})$. By Eq. 4, we have its expression as follows:

$$\begin{aligned}&\Delta D^{i,1}(L_{jwhk})\\ =&\frac{\partial}{\partial \xi^1_{jwhk}} D^i(L_{jwhk})\\ =&-\Delta q(L_{jwhk})\cdot\Delta Occ^{i,1}(L_{jwhk})\cdot\prod_{l=1}^{k-1} Occ^i(L_{jwhl})\end{aligned} \tag{12}$$

The marginal contribution of ξ^1_{jwhk} on the aggregated video distortion for user i in segment j, denoted by $\Delta\hat{D}^{i,1}_{jwhk}$, is

$$\Delta\hat{D}^{i,1}_{jwhk}=\frac{\partial}{\partial\xi^1_{jwhk}}\hat{D}^i_j=\frac{1}{|\psi^i_j|}\cdot\mathbb{1}_{T_{jwh}\in\psi^i_j}\cdot\Delta D^{i,1}(L_{jwhk}) \tag{13}$$

Taking stock from Eq. 11 to 13 above, we have the contribution of ξ^1_{jwhk} to the objective function as follows:

$$\begin{aligned}\Delta\text{AVD}^1_{jwhk}=&\frac{\partial}{\partial\xi^1_{jwhk}}\left(\frac{1}{I}\sum_{i=1}^{I}\hat{D}^i_j\right)\\ =&\frac{1}{I}\left(\Delta\hat{D}^{1,1}_{jwhk}+\sum_{i\neq 1}\Delta\hat{D}^{i,1}_{jwhk}\right)\end{aligned} \tag{14}$$

Consider two distinct ξ^i_{jwhk} and $\xi^{i'}_{jw'h'k'}$. To minimize AVD, if the marginal contributions of the downloading tasks to the objective function satisfy $\Delta\text{AVD}^i_{jwhk}<\Delta\text{AVD}^{i'}_{jw'h'k'}$, then $\xi^i_{jwhk}\geq\xi^{i'}_{jw'h'k'}$. The latter inequality implies a principle in the priority of downloading task assignments. However, the complication in the functional form of Eq. 14 and the dependence of one user's assignment on another user's video distortion make it difficult for one to go further with the mathematical derivation. Nevertheless, the details in Eq. 11 to 14 point out the following criteria that shed light on the design of the ranking approach.

1) $\Delta Occ^{i,1}(L_{jwhk})$ is independent of the value of ξ^1_{jwhk} and only varies by whether $i=1$. It simplifies the problem to a large extent by narrowing the optimization to the accounting of $\Delta Occ^{i,1}(L_{jwhk})$ in Eq. 12 to 14. Specifically, we may regard Eq. 12 as a weighted sum of the marginal qualities, Eq. 13 as an expected marginal distortion, and Eq. 14 as the group-average expected marginal distortion.
2) $\Delta Occ^{1,1}(L_{jwhk})\geq\Delta Occ^{i',1}(L_{jwhk})$ for $\forall i'\neq 1$, because $\xi^{wifi}_{jwhk}\in\{0,1\}$. This inequality indicates that the marginal gain from assigning the downloading task of a layer to its consumer is never smaller than the gain from assigning it to a device that does not need this layer and lets this device share the layer with its consumers. Therefore, every layer should be downloaded by one of its consumers.
3) Eq. 12 and 13 imply two important factors in determining the marginal gain of downloading various layers to a fixed consumer i. These determinants provide some general guidelines to the definition of the downloading priority from an individual's perspective (i.e., the device-level ranking). The heterogeneous beliefs on the relative importance of these two considerations are the driving force behind the diversified device-level ranking approaches proposed by many researchers.
   a) In Eq. 12, the last multiplicand has $\prod_{l=1}^{k-1}Occ^i(L_{jwhl})\geq\prod_{l=1}^{k'-1}Occ^i(L_{jwhl})$ for all $k<k'$. This means losing a lower-level layer causes higher video distortions. So, in the ranking approach, lower-level layers, especially the base layers, deserve higher priority.
   b) When the optimization is being solved, the mechanism attempts to minimize the video distortion of a future video segment. In this case, the future viewport has to be predicted when choosing the optimal set of layers to download, so the indicators $\mathbb{1}_{T_{jwh}\in\psi^i_j}$ in Eq. 13 are replaced by $pr(T_{jwh}\in\psi^i_j)$. Notice that $\Delta\hat{D}^{i,1}_{jwhk}$ decreases as

$pr(T_{jwh} \in \psi_j^i)$ increases. Therefore, a layer that belongs to a tile with a larger probability of being in a future viewport should have higher priority in the downloading process, and for the tiles that are more likely to be in the viewport, downloading their high-level layers are also important in minimizing video distortion. This rule points out the importance of the prediction for the future viewport in the practical approach.

4) Eq. 14 indicates that the effect of one downloading task on the group average of the video distortions is greater than its effect on the task assignee. The positive externality of the task on other users in the group is the contribution of the cooperative scheme. In any non-cooperative scheme, all ξ_{jwhk}^{wifi} would be zero, and thus, Eq. 12 and 13 would be zero whenever $i \neq 1$. This would result in a much simpler expression for $\Delta\text{AVD}_{jwhk}^1$ without any cross-user dependency. To take full advantage of this externality, a group-level centralized allocation is required to internalize the externality. This consideration motivates a group-level ranking step *after* the device-level rankings are revealed.

Similarly, we analyze the marginal contribution of various sharing task assignment to the group-level average video distortion. The marginal effect of a particular sharing task ξ_{jwhk}^{wifi} on the occupancy indicator $Occ^i(L_{jwhk})$ is defined in Eq. 15 in parallel to Eq. 11.

$$\Delta Occ^{i,\text{wifi}}(L_{jwhk}) = 1 - \xi_{jwhk}^i - \prod_{i'=1}^{I}(1 - \xi_{jwhk}^{i'}) \quad (15)$$

With similar algebraic steps as in Eq. 12 to 14, we obtain the marginal effect of ξ_{jwhk}^{wifi} on the average video distortion in Eq. 16.

Unlike the downloading task, which benefits every consumer of a layer, the marginal effect of sharing a layer on its downloader is zero, while it reduces the video distortion for other users in the group. This means the contribution of the sharing task to video distortion differs from the contribution of downloading it by the utility of the layer to its downloader. This implies that there should be a ranking system for sharing tasks, and it can be modified from the ranking for downloading tasks by eliminating the effect of each layer on its first owner in the group.

$$\begin{aligned}&\Delta\text{AVD}_{jwhk}^{\text{wifi}} \\ &= -\frac{\Delta q(L_{jwhk})}{I}\sum_{i=1}^{I}\left[\frac{\mathbb{1}_{T_{jwh}\in\psi_j^i}}{|\psi_j^i|}\cdot\Delta Occ^{i,\text{wifi}}(L_{jwhk})\right. \\ &\left.\cdot\prod_{l=1}^{k-1}Occ^i(L_{jwhl})\right]\end{aligned} \quad (16)$$

B. Overview of the Ranking Approach

Inspired by the inferences we find from the mathematical derivation, the ranking approach includes three steps:

Step 1: The server, having collected the video-watching information from all devices in the group, predicts the future demand for various layers and then ranks all layers for each device separately, incorporating the prediction and its possible errors.

Step 2: The server compiles the device-level rankings into a group-level ranking, discussed in detail in subsections IV-C and IV-D.

Step 3: The ranking is used to determine the set of SVC layers being downloaded and shared by the group, by adding layers from the ranking list sequentially into the set for transmission, until the sum of the bandwidths required for the downloading task fills the capacity of the cellular network on each device, and the sum of the bandwidth required for the sharing task fills the capacity of the MANET.

The bandwidth restriction and fluctuation is incorporated instead of neglected by the ranking approach. First, the ranking approach ensures the order of priority and the SVC layering provides flexibility to guarantee that the network resources needed to fulfill the downloading assignments do not exceed the bandwidth but are close enough to the bound. Second, the cooperative scheme assigns tasks dynamically with respect to each device's network ability to avoid big transmission loss due to bandwidth fluctuations. The small sizes of the task assignments are enabled by SVC layering.

Now, we recap the mathematical evidence we characterized from the theoretical optimal solution and corresponding components in the ranking approach to restate their qualitative equivalence. The capacity constraints listed in Eq. 10 are met in step 3 of the ranking approach described above and are thus irrelevant in the implementation of the scheme because the heuristic approach ranks all the SVC layers into an ordered list and downloads as much as the actual bandwidth allows. Hence, the approach uses the bandwidth fully while not exceeding it. The task duplication constraint is met by the server assigning the tasks using the group-level ranking from a collective perspective. The objective function of the optimization problem is a negative function of the quality of the effective layers, and the accounting of device-level distortions implies that this objective function is also a positive function of the number of consumers for each given SVC layer. However, the device-level ranking of the above approach prioritizes the layers based on the expected gain from their qualities. The collective ranking is obtained through the unweighted sum of the reciprocals of the squares, so it guarantees that the ranking gives credit to high quality as well as the number of consumers of various layers. Thus, the general pattern of this ranking approach is the same as the qualitative characteristics of the objective function.

C. Device-Level Ranking Mechanisms

In the spirit of the criteria mentioned in Subsection IV-A, this subsection proposes three candidates for the device-level ranking mechanism. In these mechanisms, our main objectives include the quality of the tiles, as well as their accessibility, echoing the functional form expressed in Eq. 13.

1) Prediction-based Zare Mechanism (Pred-Zare): The first mechanism consists of two features. It uses weighted linear regression (WLR) as suggested by Qian et al. [9] to predict the direction of the viewport in the next time period. It then takes

$$A^{-1} = \begin{bmatrix} -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\beta & -\sin\alpha\cos\beta\sin\gamma - \cos\alpha\sin\beta & -\sin\alpha\cos\gamma \\ \sin\beta\cos\gamma & \cos\beta\cos\gamma & -\sin\gamma \\ \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\beta & \cos\alpha\cos\beta\sin\gamma - \sin\alpha\sin\beta & \cos\alpha\cos\gamma \end{bmatrix} \quad (17)$$

the spirit of Zare et al.'s binary approach [8] to try to download the tiles in the predicted area with the highest quality and those outside of the area with basic quality if bandwidth resources allow. It acquires the tiles in the predicted effective area from base to enhancement SVC layers. In each group of layers with the same priority in terms of the likelihood of visibility and the SVC layer level, these layers are acquired from high to low marginal quality gain (i.e., $\Delta q(\cdot)$). Note that this prediction method implies a probability of 1 for the predicted area and 0 for all other tiles, which means that the prediction can only underestimate the probability of demanding a tile outside of the predicted area but never overestimate the probability of demanding any tile. This concern motivates our ranking mechanism to consider prediction *errors* instead of *inaccuracy*.

2) Zigzag Mechanism: The second mechanism ranks the video data in a zigzag manner. It is a continuous version of the previous mechanism. In this mechanism, priority cutoffs between highly likely useful layers and all other layers are less dramatic. Using this mechanism avoids the prediction error that we pointed out in Subsection IV-C1 because it gradually expands from the current viewport rather than making a directional prediction. To quantify the likelihood that a certain layer is or will be demanded, we compose a process to measure the distance from the center of the projected area of the viewport on the sphere to the center of various tiles on the sphere. Then, we compose a ranking order combining the distance measures and marginal layer qualities, and this order has a zigzag pattern as visualized in Fig. 5.

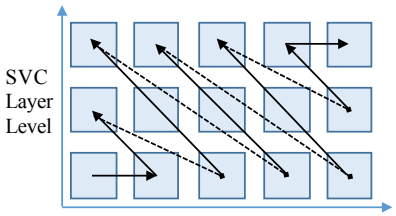


Fig. 5. Zigzag Device-level Ranking Mechanism ($\eta = 0.5$)

The video that the mechanism is processing is a 360-degree panoramic video, so the same video tile can be represented in multiple coordinate systems. The video is stored on pixel coordinates in the source file, and the viewport is labeled on the spherical coordinates. To measure the distance between points on the viewport and points of the video tiles, we project both inputs to three-dimensional Cartesian coordinates. Next, we find the Euclidean distance between the two points to define $\theta(P, Q)$ as our measure for the viewport's travel distance between two points on the sphere as follows:

$$\theta(P, Q) = 2 \cdot arcsin(\frac{\|P - Q\|}{2 \cdot \|P\|}) \quad (18)$$

with $\|\cdot\|$ being the Euclidean distance.

The edges of the viewport may not be perfectly vertical or horizontal in relation to the axes of the video tiles. When they are not, the relative positions of the viewport versus the video tiles can be represented by a rotation matrix that tilts a horizontal and vertical viewport or by another rotation matrix that spins the spherical video tiles to the desired angle. Note that these matrices are the inverse of each other, and the outcomes of these two procedures are the same. The Android platform adopts the latter matrix A [22], while we import the matrix and invert it to A^{-1} to adopt the former. The inverted matrix A^{-1} is presented in Eq. 17, in which (α, β, γ) are (roll, yaw, pitch) in typical human head movements. We believe that the first procedure is computationally more efficient as it only processes the viewport, which is a small subset of all tiles.

We incorporate the travel distance function and the inverse-rotation matrix proposed above and define the distance between a tile and the viewport as follows:

$$dist^i(L_{jwhk}) = \theta(V \cdot A^{-1}, L_{center}) \quad (19)$$

in which V is the center of the viewport, and L_{center} is the center of L_{jwhk}. Note that both points and A^{-1} are on Cartesian coordinates, and neither point requires rotation.

The SVC layers of tiles are then grouped into bins by their distance from the viewport, and the cutoffs between the bins are multiples of a fixed width κ. Equivalently, we define that layer L_{jwhk} belongs to the bin b (i.e., $bin(L_{jwhk}) = b$) if and only if Eq. 20 is satisfied.

$$\kappa \cdot b \leq dist^i(L_{jwhk}) < \kappa \cdot (b + 1) \quad (20)$$

Next, we use the bin number $bin(L_{jwhk})$, layer level k, and travel distance provided by each tile $dist^i(L_{jwhk})$ to construct a function to represent the ranking order. We propose the following functional form for the device-level ranking score:

$$\begin{aligned} rank^i(L_{jwhk}) =& exp(2 \cdot (\eta_j^i \cdot k + (1 - \eta_j^i) \cdot bin(L_{jwhk})) \\ &+ exp(-\frac{1}{1+k})) + exp(-\frac{1}{dist^i(L_{jwhk})}) \end{aligned} \quad (21)$$

In the above equation, we use η, a parameter ranging between 0 and 1, to represent the relative importance of a higher level layer versus base layers of additional tiles. It is a casual measure for the excess kurtosis of the quality distribution across tiles and layers. Specifically, as shown in Eq. 22, η is the ratio between the total quality of all layers of all tiles in the current viewport and the total marginal quality of the base layers of all tiles on the sphere. This ratio is usually in the range of $[0.3, 0.6]$ in real-world experiments.

$$\eta_j^i = \frac{\sum_{T_{jwh} \in \psi_j^i} \sum_{k=1}^{K} \Delta q(L_{jwhk})}{\sum_{\forall w,h} \Delta q(L_{jwh1})} \quad (22)$$

Note that a layer with a lower ranking score has higher priority. The intuition for the ranking function is that we offer the highest priority to the sum of the bin number and layer level, then the second-order priority goes to the layer level itself, and then within each $(bin(L_{jwhk}), k)$ combination, the

SVC layers are ranked by the additional quality they yield. As illustrated in Fig. 5, if we compute the ranking score of each layer and draw arrows in the order of the rankings of all layers, the line chart is a zigzag type of plot.

3) Prediction-based Zigzag Mechanism (Pred-Zigzag): The third mechanism is a combination of the two mechanisms described in Subsections IV-C1 and IV-C2, which builds the Zigzag ranking mechanism on the prediction method. The composite mechanism centers from the predicted future viewport and expands its downloading area in a zigzag manner. This offers a variation of the basic Zigzag mechanism.

This mechanism is supposed to perform better than the Zigzag mechanism if the prediction is accurate and perform better than the Pred-Zare mechanism if the prediction often has small biases. However, as we will discuss in Subsection V-D, the prediction method does not work well for a relatively long future (3 seconds) and causes both Zigzag and Zare mechanisms to work better when centering around the current viewports than around the predicted future viewports.

D. Group-Level Ranking Mechanism

Anyone with the ranking mechanisms provided above could produce an *order index* of the ranking list generated for each device. This order index is then reported to the server, which compiles the order indices and composes the collective ranking for this group. The essential task for us is to construct a function that uses the device-level indices of a certain layer to produce a group-level ranking score for this layer. According to Subsection IV-A, this function should consider the value-added by each layer from the group planner's perspective. Let us denote the device-level order index by $\{index^i(L_{jwhk})\}$; thus, the group-level ranking score of L_{jwhk} is defined by the following function:

$$rank_g(L_{jwhk}) = \sum_{i=1}^{I} \left[index^i(L_{jwhk})\right]^{-2} \tag{23}$$

Remark. *The functional form of this group-level ranking score comes from the consideration that we want to give very high priority to a layer with a very small (upfront) ranking index from at least one of its consumers, even if its other consumers put it in low priorities. For example, if layer L_1 is ranked 1^{st} by one device and 10^{th} by another, while layer L_2 is ranked 5^{th} and 6^{th}, we want the above function to give a smaller ranking index to L_1 than L_2. This preference directs us to a sum of concave functions of the device-level ranking indices. Furthermore, for the sake of computational simplicity, we use the inverse of the square of the device-level ranking index as the specific form for the concave function.*

The group-level ranking scores are then sorted descendingly to construct the group-level ranking list for all the layers of this video source file.

E. Generation of Task Sets

The server follows the group-level ranking list to allocate the downloading tasks accordingly under the bandwidth constraints in a dynamic manner to keep the number of pending tasks per device in the range of 1 and the threshold C. The server waits until at least one device falls short on pending downloading tasks to start assigning the foremost unassigned layer in the group-level ranking list. In such a scenario, it assigns the layer to the device with the topmost device-level order index for this layer from the set of devices with pending tasks below the threshold. The above procedures are summarized in Algorithm 1.

Algorithm 1 Algorithm for Downloading Tasks Assignment

1: *Initialization*: $task_i =$ empty list for all i
2: **while** In the time block of downloading a certain video segment S_j **do**
3: **if** $\min\{|task_i|\} \leq 1$ **then**
4: **repeat**
5: Find the foremost unassigned layer $L^* = \underset{L_{jwhk}}{\operatorname{argmax}} \left\{ rank_g(L_{jwhk}) \text{ s.t. } \sum_{i=1}^{I} \xi^i_{jwhk} = 0 \right\}$
6: Randomly pick one user i^* from the set $\underset{i}{\operatorname{argmin}} \left\{ index^i(L^*) \text{ s.t. } |task_i| < C \right\}$ and let $\xi^{i^*}_{jwhk} = 1$.
7: **until** $\min\{|task_i|\} \geq C$
8: **end if**
9: **end while**

On the sharing part of the task assignments, downloaded SVC layers are re-ranked within the group and shared by the downloader (owner) in the descending order of the sharing ranking score $rank_s$ until the local MANET's bandwidth is fully utilized. The $rank_s$ score is defined as follows:

$$rank_s(L_{jwhk}) = rank_g(L_{jwhk}) - \left[index^{owner}(L_{jwhk})\right]^{-2} \tag{24}$$

F. Complexity Analysis and Performance Upper Bound

In this subsection, we analyze the complexity performance of our mechanism to find its upper bound. Our mechanism consists of downloading and sharing tasks. The downloading task itself consists of an Algorithm 1 and the ranking mechanism, with the latter done for device- and group-level ranking.

The level of complexity mainly depends on two measures: the number of users, denoted by I, and the total number of layers in a particular video segment, denoted by $n = W \cdot H \cdot K$. In the device-level ranking of each of the I devices, in which we analyze the Zigzag ranking mechanism as an example, Eq. 21 and 22 are evaluated n times, and the total computational complexity is $O(n)$. The mechanism then sorts the ranking scores to obtain the device-level index. In the group-level ranking mechanism, we evaluate Eq. 23 n times with $O(In)$ and sort the values with $O(n \log n)$. Therefore, we can see that the ranking mechanisms together have an upper bound of $O(I \cdot (n + n \log n) + In + n \log n) = O(In \log n)$. Next, Algorithm 1 constantly generates new downloading task assignments throughout the time block. Lines 3 and 7 constitute an if condition with a complexity of $O(I)$ and define an inner loop. Each iteration of the inner loop generates an assignment that can be handled instantly. The complexity of the algorithm is represented by the complexity of one independent iteration of the inner loop. Line 5 uses the group-level ranking outcomes, so it does not require additional computation. Line 6 takes the time complexity of $O(In)$.

Thus, we can also see that Algorithm 1 has a total complexity of $O(In)$.

As a result, the upper bound of our scheme on the server's side is $O(In \log n + In) = O(In \log n)$. Note that the practical magnitudes of n and I are usually small, ensuring the efficiency of our heuristic scheme. Meanwhile, mobile devices generate sharing tasks and evaluate the share ranking scores with an upper bound of $O(n)$ and sort them with $O(n \log n)$.

V. Experiments

To compare the performance of our scheme with other schemes, we built a prototype on the Android platform for real-world experiments. Subsection V-A describes the environment in which the experiments are performed. Subsection V-B presents the performance of our proposed scheme relative to existing schemes in the literature. Subsection V-C and V-D decompose the contribution of our scheme into the value added by cooperation and the device-level ranking mechanism, respectively. Subsection V-E discusses the smoothness of the video quality. Subsection V-F performs robustness checks by varying the Quantization Parameter (QP) settings and video quality measurements. Subsection V-G presents the performance with an alternative QoE measure. Finally, Subsection V-H discusses the practical issues.

A. Experimental Setup

The experiments use a 360-degree panoramic video[1] from Youtube [23] with a resolution of 3840×2048 and a frame rate of 30 fps. This video is divided into 1-second video segments, each of which is cropped into 8×4 tiles, and thus, each tile has a resolution of 480×512. Each tile is encoded using H.264/SVC reference software JSVM 9.19.15 [24] into 3 layers following the quality adaptation. An SVC Demux/Mux tool [25] is used to de-multiplex and merge the SVC layers. In the Zigzag device-level ranking mechanisms, the bin width is set to be $\kappa = 10°$.

Note that QP values are fixed to (40, 28, 20) and the video quality is measured by the PSNR method in Subsections V-B through V-E to compare the performance of our scheme to other existing schemes. Then, additional experiments using different QP and measurement settings are performed to show the robustness of the performance of our scheme. In these experiments, the QP values have been chosen by considering the variation in their corresponding data amount, the marginal quality of the layers, and the overall quality measure of the video file. The variation is used to illustrate that our proposed schemes perform better for different QP and quality measure environments. We implemented the proposed scheme on the Android platform of the Nexus 5 (hammerhead) and used the Cyanogen Mod 11 as the operating system. The viewport size of the smartphone was 120 horizontal with 90 vertical degrees (i.e., a resolution of 1280×1024). The operator of the cellular path using LTE was China Unicom. The server was deployed on the Elastic Compute Service (ECS) of Aliyun [26] equipped with an eight-core 2.6 GHz CPU and 8 GB RAM.

The reference schemes that we compare the performance of our proposed scheme with are:

- Zare et al.'s [8] scheme, which downloads the video with the highest quality in the current viewport and the base layer for all other tiles;
- Qian et al.'s [9] scheme, which predicts the viewport using WLR and only downloads the layers of tiles in the predicted viewport;
- The Pyramid scheme [13], which predicts the viewport in the same way as Qian and assigns tiles' downloading priority based on their distance from the predicted viewport;
- The 360ProbDASH scheme [10], which predicts the probability distribution of the future viewport, computes the probability of each tile being watched, and downloads the video tiles accordingly.

To make the procedures in our schemes comparable with these reference schemes, a series of minor modifications are made to adopt the SVC layering concept to the reference schemes. We first project the low-, medium-, and high-quality versions of the video in AVC schemes onto corresponding cumulations of the SVC layers. While the SVC layering provides additional flexibility, we respect the specification of the reference schemes regarding the downloading set and priority, as well as the correspondence between the AVC versions and the SVC layers. We then ask the schemes to download the SVC layers in the order of the dependency of these layers. This requirement enables video decoding with certainty and thus allows for the strict timeline on the reference schemes after we eliminate the roles played by video jitter and playback buffering. When the downloading deadline hits, we ask the devices to decode and render all successfully received SVC layers.

In this section, every experiment consists of 10 repetitions[2]. In each repetition, we test the schemes on three smartphones placed at the same location, and we let them display different portions of the same 360-degree panoramic video at the same time. This way of using three mutually independent traces mimic the human behavior of three different people when watching the same 360-degree video. While running the experiments in a real-world environment, we let the smartphones download the video files in real-time and keep record of which tiles and layers they have obtained to see how the system performs on downloading the video files with respect to the traces. The average video distortion is computed by comparing the video quality of the source file and real-time downloaded video quality among the tiles being watched.

B. Overall Performance

In this subsection, we compare the overall performance of our proposed scheme with the performance of the reference

[1]The video source we use is a 91-second long, 360-degree time-lapse video of a New York City street view. It is a good resource for experiments that emphasizes different moving viewports across users, as its favorite points of the video change over time and are not unique.

[2]The experiments use trace data from [27]. For each repetition, we randomly select 3 entries from the set of all 58 traces. The 10 repetitions involve the following traces: (dae1, 2681, 835a), (c8e3, c0d3, 1f0a), (b433, 6e09, 2414), (471f, a0e6, 9e90), (dd59, 3c32, f3d3), (471f, d49b, ea4a), (7092, 5bac, 944e), (2414, ea4a, f3d3), (f6a6, d49b, 1ce0), (6890, 86db, aa17).

schemes to show the pooled effect of cooperative video delivery and the device-level ranking mechanism on improving the video quality of existing techniques.

Fig. 6 presents the empirical cumulative distribution function (CDF) of the AVD derived from the experiments on various schemes[3]. The empirical CDF curves are aggregations of the outputs from 614 rounds of optimization from all 10 repetitions. The optimization problems are solved independently with no serial correlation, so the group-level AVD can be considered a statistic following the same distribution. This independence justifies the construction of the CDF.

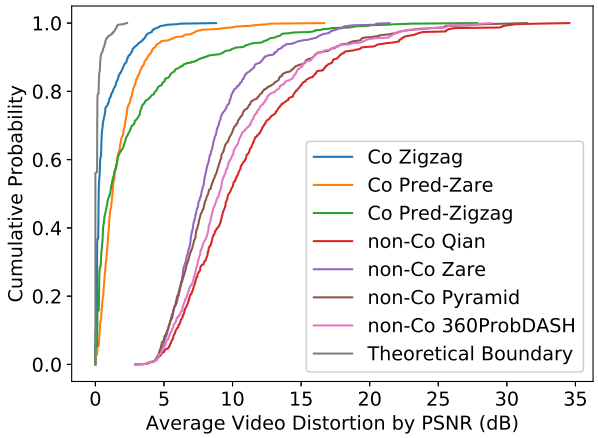


Fig. 6. CDF of AVD by Schemes using Trace Data

Our cooperative schemes perform significantly and uniformly better than the non-cooperative ones by generating less distortion and better video quality. It benefits from cooperation to expand group level bandwidth, seeking potential common demands to avoid duplication and the device-level ranking design to accommodate prediction errors. Comparing the three device-level ranking mechanisms we proposed, the slightly poor performance of the three mechanisms involving prediction indicates that the WLR prediction method overestimates human head movements in general. The cooperative scheme using the Zigzag device-level ranking mechanism improves significantly upon previously existing schemes. Furthermore, since our scheme is more effective than the non-cooperative tile-based schemes, the cooperative tile-based scheme we proposed is better than the non-viewport-adaptive schemes (e.g., current Youtube 360-degree videos).

In addition, we find the ex-post optimality as the theoretical boundary of the constrained minimization problem defined in Eq. 10. Specifically, we first record real-time transmission information, including bandwidth variations and network constraints. We also assume all future viewports are known. Then, we use the dynamic programming method to find the minimal average video distortion defined by the optimal downloading and sharing tasks. Note that this solution requires an unrealistically always-perfect guesser and an impractical solving procedure with an exponential computational complexity, which is why this can only be obtained offline as a theoretical solution. The schemes proposed in this paper are all much closer to the theoretical optimum than any of the reference schemes. This shows that our proposed schemes have eliminated most of the video distortion that they possibly could with high probability.

[3]In the legend of this figure and all of the subsequent figures, "Co" is the abbreviation of "cooperative".

C. Value Added by the Cooperation Manner

We first justify the contribution of cooperation to the performance of the proposed scheme. To apply cooperation on the referenced mechanisms, we treat them as four types of device-level ranking mechanisms and fit them in the cooperative architecture we proposed in this paper. A cooperative architecture requires 3τ (3 seconds) of processing time, so the prediction lags we impose on the device-level Qian, Pyramid, and 360ProbDASH approaches are extended from 1 second to 3 seconds. The statistic we present in Fig. 7 is the percentage decrease in AVD that cooperation brings to each device-level ranking mechanism (i.e., $(\text{AVD}_{\text{noncoop}} - \text{AVD}_{\text{coop}})/\text{AVD}_{\text{noncoop}}$).

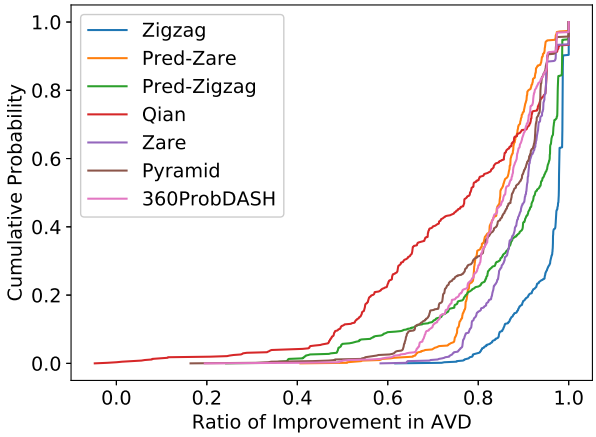


Fig. 7. Fraction of Improvement in AVD due to Cooperation

The improvements in the schemes' performances are significantly positive for all device-level ranking mechanisms[4]. In particular, cooperation reduces AVD by at least 40% almost surely, showing that the good performance of the schemes proposed in this paper benefits from group cooperation.

D. Value Added by the Device-Level Ranking Mechanisms

To compare the degree of improvement provided by the device-level ranking mechanisms proposed in this paper, we applied different ranking mechanisms, including the reference schemes with minor transformations, to the same cooperative architecture built in this paper. Fig. 8 presents the empirical CDF of the AVD of the cooperative schemes with various device-level ranking mechanisms.

Among all seven device-level ranking mechanisms, the performance of the Zigzag mechanism proposed in this paper outperforms the other alternatives with a probability of almost one. The position of Zare in Fig. 8 is much better than its position in Fig. 6, as its cooperative version benefits from the efficient allocation of bandwidth resources. Viewport prediction power decreases sharply as the prediction time

[4]Except for Qian's approach, which does not benefit from cooperation with a small probability. The limited improvement is caused by prolonged prediction lag and the approach's specification, which downloads nothing outside of its predicted viewport.

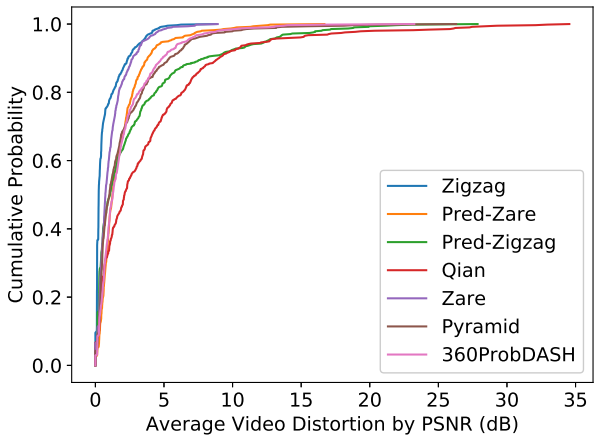


Fig. 8. CDF of AVD of Cooperative Schemes by Device-level Ranking Mechanisms

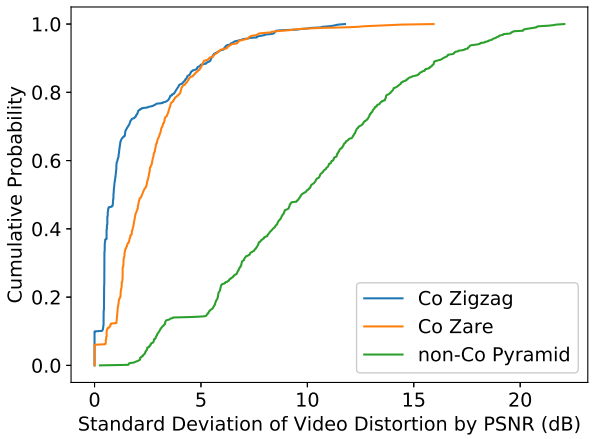


Fig. 9. CDF of Standard Deviation of Video Distortion

is prolonged to 3 seconds to accommodate the cooperative schemes. The enlarged prediction error puts Qian in a poor performance state and makes Zare and Zigzag outperform Pred-Zare and Pred-Zigzag, respectively.

While the six ranking mechanisms that exhaust the bandwidth resource for transmission perform similarly, Qian's has the lowest performance among all mechanisms. The potential improvement in the performance of Qian by using more resources is bounded above by Pyramid and 360ProbDASH, which shows a modest advance in users' QoE. Among the six approaches that consume the same amount of resources, their differentials come from two aspects. First, the advantage of using layer-level ranking instead of tile-level ranking is revealed in Zigzag versus Pyramid and 360ProbDASH. Second, the advantage of using continuous ranking instead of concentric discrete ranking is shown in Zigzag versus Zare and Pred-Zigzag versus Pred-Zare.

E. Variation in the Tile-Level Video Quality

In addition to evaluating the average video distortion of a group of users over the period of the video as in Subsection V-B, we consider the variation in the tile-level video quality over all tiles on the viewport as a measurement for the smoothness of the video quality on the screen. This is also an important factor affecting consumers' QoE as they are sensitive to the variation of video quality across different areas of the viewport [10], [28]. A smaller standard deviation in tile-level video distortions is preferred, holding their average constant.

The CDF of the distribution of this standard deviation for selected schemes is presented in Fig. 9, with the standard deviation defined in Eq. 25.

$$\sigma_j = \frac{1}{I}\sum_{i=1}^{I} std\left(\left\{\sum_{k=1}^{K} D^i(L_{jwhk})\right\}_{T_{jwh}\in\psi_j^i}\right) \tag{25}$$

The figure shows that the cooperative schemes perform well at stabilizing users' QoE, and the Zigzag mechanism takes advantage of the continuity ingredient in the device- and group-level ranking process to smooth the video's quality.

F. Robustness Checks

We now show that the superior performance of our scheme is independent of the choice of video quality measures and QP settings of the video sources. The marginal quality contribution of the SVC layers of the video file can be measured by PSNR and SSIM, and the QP settings affect the distribution of data and marginal quality through the encoding process. The optimization problem depends on these two parameters since the distribution of marginal quality affects the value of η (Eq. 22), and the data amounts of various layers affect the downloading tasks under the capacity constraints.

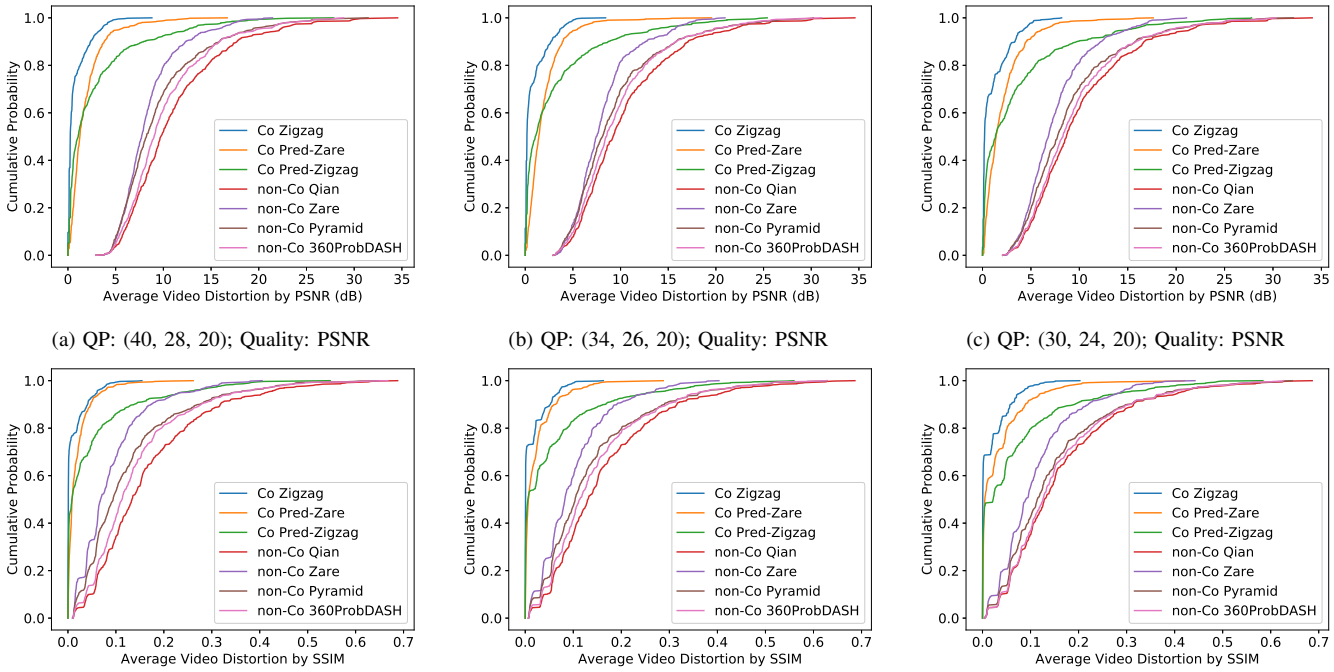
Table II summarizes the three different environments in which we tested our schemes. They vary by the QP settings and, thus, by the data amounts. The settings, along with the choice of video quality measures, determine the marginal qualities associated with each layer. Fig. 10 shows the performance of the three schemes we proposed versus the previously existing schemes under three QP settings by two quality measures. Although the details vary across specifications, the Zigzag approach performs the best, showing the robustness of our schemes' good performance. We also ran experiments with various video sources, tile numbers, and viewport sizes. As the outcomes of these experiments are qualitatively the same as the main experiment as shown in Fig. 6, we do not present these results in detail.

G. Alternative QoE Measurement

We also evaluate the performance of our proposed schemes and the reference schemes using a composite QoE measurement metric proposed by Yin et al. [29]. The measurement sums video quality, quality variation, rebuffer time, and initial startup delay. We adopt this framework and make the following specifications. A user can only view one portion of a 360-degree panoramic video at any given time. We define the visible bitrate of user i on video segment j as the sum of the bitrate of all tiles that user i views in segment j. Its definition is presented in Eq. 26.

$$\text{Rate}_j^i = \sum_{\forall w,h,k}\sum_{T_{jwh}\in\psi_j^i}\left(\delta(L_{jwhk})\cdot\prod_{l=1}^{k} Occ^i(L_{jwhl})\right) \tag{26}$$

Then, we formulate the four summands in the composite QoE measurement. Video quality is the sum of visible bitrates



(a) QP: (40, 28, 20); Quality: PSNR (b) QP: (34, 26, 20); Quality: PSNR (c) QP: (30, 24, 20); Quality: PSNR

(d) QP: (40, 28, 20); Quality: SSIM (e) QP: (34, 26, 20); Quality: SSIM (f) QP: (30, 24, 20); Quality: SSIM

Fig. 10. CDF of AVD in Six Experiment Environments with Various Quality Measures and QP Settings (Note that Fig. 10a is the same as Fig. 6.)

TABLE II
DISTRIBUTION OF DATA AMOUNT AND VIDEO QUALITY OF DIFFERENT EXPERIMENT ENVIRONMENTS

	(1)	(2)	(3)
QP	(40, 28, 20)	(34, 26, 20)	(30, 24, 20)
Data (Mbps)	3.4728 18.4046 47.1577	7.1792 21.1998 46.0552	11.3301 26.1751 46.6820
Quality (PSNR)	33.4870 41.2672 47.0040	37.1349 42.6841 47.0753	39.7888 44.2175 47.0599
Quality (SSIM)	0.8737 0.9747 0.9923	0.9374 0.9812 0.9924	0.9644 0.9864 0.9923

Notes: (1) The three numbers in each cell of the table correspond to the statistics of the base layer, enhancement layer 1, and enhancement layer 2, respectively. (2) The data amounts are cumulative in terms of layers. The data amount for the k^{th} layer is computed by averaging the total data amounts up to this layer for the entire video file over time segments (i.e., $\frac{1}{J}\sum_{\forall j,w,h}\sum_{l\le k}\delta(L_{jwhl})$). (3) The quality measure is also cumulative. The quality of the k^{th} layer is the two-dimensional (temporal and spatial) average of the quality of the video up to this layer, over all time segments and all layers of the video file (i.e., $\frac{1}{J\cdot W\cdot H}\sum_{\forall j,w,h} q(L_{jwhk})$).

across all video segments. Quality variation is the sum of absolute values of visible bitrates of each consecutive pair of video segments. The rebuffer time is zero for all schemes discussed in this paper because the strict time constraint in transmission prevents video jitter. Startup delay is the time required to download, share and decode the SVC layers of the first video segment. Note that the startup delay length is invariant for each scheme and equals 3τ for all cooperative schemes in the experiment. We multiply the startup delay by a coefficient to scale it to the same magnitude as video quality and quality variation. The scaling coefficient equals the maximum of visible bitrates among all users and video segments. Lastly, we define the composite QoE of a scheme involving multiple users as the average of the composite QoE over this set of users, which is defined in Eq. 27.

In addition, we define the theoretical QoE as the QoE of a cooperative scheme with perfect prediction. This theoretical QoE measurement is used to normalize the QoE of various schemes [29].

$$\begin{aligned}\mathrm{QoE}_1^J =& \frac{1}{I}\sum_{i=1}^{I}\left(\sum_{j=1}^{J}\mathrm{Rate}_j^i - \sum_{j=1}^{J-1}\left|\mathrm{Rate}_{j+1}^i - \mathrm{Rate}_j^i\right|\right) \\ & - 0 - \max_{\forall i,j}(\mathrm{Rate}_j^i)\cdot 3\tau\end{aligned} \quad (27)$$

Fig. 11 provides the CDF of the normalized QoE. It shows that the cooperative schemes perform almost as good as the theoretical boundary of the QoE on this measure. They also significantly outperform the non-cooperative schemes. The cooperative Zigzag approach proposed in this paper generates the highest QoE of all schemes.

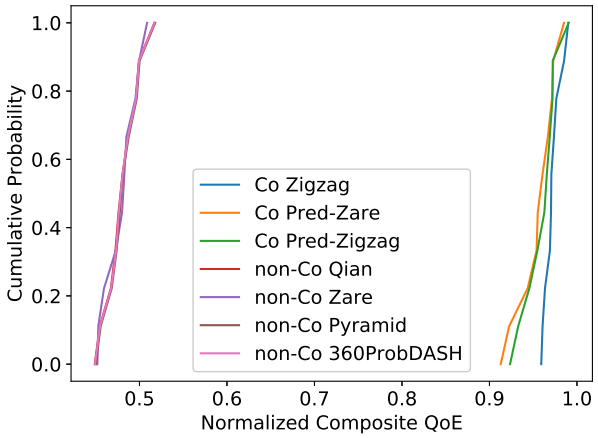


Fig. 11. CDF of Normalized Composite QoE

H. Practical Issues

When implementing our scheme, the most likely practical issue one would encounter is the implantation of the broadcast

technique via MANET on existing smartphones. Only a few manufacturers natively support this technique on their products. In the academic context, researchers could modify the Linux kernel to activate the broadcast technique on Android smartphones. In commercial use, manufacturers could provide the technology to consumers by offering the fully functional version of the Linux kernel.

The lack of native support for SVC on existing smartphones is another roadblock in the practical use of our schemes. To overcome this problem, SVC-based schemes rely on software to decode the video files. In doing so, two issues arise. First, the decoding process is considerably slow and, thus, our schemes reserve a time block for the procedure. Second, it is energy demanding, which is unavoidable at the moment. Hence, the commercial implementation of our schemes has to wait for manufacturing advancements in the hardware of smartphones.

VI. Conclusion

This paper solved a QoE maximization problem in which multiple devices in close proximity are simultaneously displaying the same 360-degree panoramic video, although from different angles. Our solution is a cooperative scheme that identifies and takes advantage of the common portion of the users' video demands. We developed several device-level ranking mechanisms to translate individual-specific video demands into a priority preference in downloading tasks. We formulated the group-level QoE maximization problem as an AVD minimization problem under feasibility constraints to provide a practical solution. In the practical scheme, we transformed the individual-level ranking into an aggregated group-level ranking to allocate resources from a collective perspective. The real-world implementation of our proposed scheme shows that in terms of generating lower levels of AVD, it performs significantly better than existing non-cooperative schemes, and the individual-level ranking mechanism contributes significantly to this improvement.

Acknowledgment

The authors would like to thank the anonymous reviewers for many valuable comments. This work is partly supported by the China Scholarship Council.

References

[1] Digi-Capital, "Augmented/virtual reality revenue forecast revised to hit $120 billion by 2020," 2016. [Online]. Available: http://goo.gl/Lxf4Sy

[2] Google, "Google Cardboard," 2017. [Online]. Available: https://vr.google.com/cardboard/index.html

[3] Samsung, "Samsung Gear VR," 2017. [Online]. Available: http://www.samsung.com/global/galaxy/gear-vr/

[4] E. W. Weisstein, "Equirectangular projection," 2017. [Online]. Available: http://mathworld.wolfram.com/EquirectangularProjection.html

[5] W. Mason, "VR HMD Roundup: Technical Specs," 2016. [Online]. Available: http://uploadvr.com/vr-hmd-specs/

[6] S. Hollister, "Youtube's ready to blow your mind with 360-degree videos," 2015. [Online]. Available: https://gizmodo.com/youtubes-ready-to-blow-your-mind-with-360-degree-videos-1690989402

[7] E. Kuzyakov and D. Pio, "Next-generation video encoding techniques for 360 video and VR," 2016. [Online]. Available: https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/

[8] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 601–605.

[9] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 1–6.

[10] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360probdash: Improving qoe of 360 video streaming using tile-based http adaptive streaming," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 315–323.

[11] Z. Xu, X. Zhang, K. Zhang, and Z. Guo, "Probabilistic viewport adaptive streaming for 360-degree videos," in *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*. IEEE, 2018, pp. 1–5.

[12] D. V. Nguyen, H. T. Tran, A. T. Pham, and T. C. Thang, "A new adaptation approach for viewport-adaptive 360-degree video streaming," in *Multimedia (ISM), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 38–44.

[13] H. Ahmadi, O. Eltobgy, and M. Hefeeda, "Adaptive multicast streaming of virtual reality content to mobile users," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. ACM, 2017, pp. 170–178.

[14] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "Optile: Toward optimal tiling in 360-degree video streaming," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 708–716.

[15] B. Zhang, Z. Liu, S.-H. G. Chan, and G. Cheung, "Collaborative wireless freeview video streaming with network coding," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 521–536, 2016.

[16] A. Le, L. Keller, H. Seferoglu, B. Cici, C. Fragouli, and A. Markopoulou, "Microcast: Cooperative video streaming using cellular and local connections," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2983–2999, oct 2016.

[17] X. Zhang, J. Liang, and L. Zhang, "Delay-constrained streaming in hybrid cellular and cooperative ad hoc networks," *Computer Communications*, vol. 118, pp. 205 – 216, 2018.

[18] Y. Zhang, C. Li, and L. Sun, "Decomod: collaborative dash with download enhancing based on multiple mobile devices cooperation," in *Proceedings of the 5th ACM Multimedia Systems Conference*. ACM, 2014, pp. 160–163.

[19] L. Zhang, X. Zhang, K. Qu, L. Ren, J. Deng, and K. Zhu, "Green and cooperative dash in wireless d2d networks," *Wireless Personal Communications*, vol. 84, no. 3, pp. 1797–1816, 2015.

[20] N. Abedini, S. Sampath, R. Bhattacharyya, S. Paul, and S. Shakkottai, "Realtime streaming with guaranteed qos over wireless d2d networks," in *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2013, pp. 197–206.

[21] E. Yaacoub, F. Filali, and A. Abu-Dayya, "Qoe enhancement of svc video streaming over vehicular networks using cooperative lte/802.11 p communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 37–49, 2015.

[22] Android Developers, "SensorManager," 2017. [Online]. Available: https://developer.android.com/reference/android/hardware/SensorManager.html#getRotationMatrix(float[],float[],float[],float[])

[23] BOLD WORLDWIDE, "NYC 360 Time-Lapse (360 Video)," 2017. [Online]. Available: https://www.youtube.com/watch?v=CIw8R8thnm8

[24] Joint Video Team, "Joint Scalable Video Model Software, Version 9.19.15," document JVT-AF013, 2009.

[25] C. Kreuzberger, D. Posch, and H. Hellwagner, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over http," in *Proceedings of the 6th ACM Multimedia Systems Conference*. ACM, 2015, pp. 213–218.

[26] Aliyun, "Elastic Compute Service," 2017. [Online]. Available: https://cn.aliyun.com/product/ecs

[27] X. Corbillon, F. De Simone, and G. Simon, "360-degree video head movement dataset," in *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 2017, pp. 199–204.

[28] H. Wang, V.-T. Nguyen, W. T. Ooi, and M. C. Chan, "Mixing tile resolutions in tiled video: A perceptual quality assessment," in *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*. ACM, 2014, p. 25.

[29] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 325–338.