Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 12, No. 01, March 2026.

VEGA-360: VIEWPORT-AWARE HIERARCHICAL GROUPED ALLOCATION FOR MULTI-LAYER 360° VIDEO STREAMING

Phi Dinh Huynh and Nguyen Viet Hung

(Received: 26-Nov.-2025, Revised: 13-Jan.-2026, Accepted: 16-Jan.-2026)

ABSTRACT

Delivering multi-layer, tiled 360° video to multiple wireless users is challenging due to limited radio resources, heterogeneous channel conditions and a strong viewport-dependent quality of experience (QoE). To address this problem, we propose Viewport-Enhanced Grouped Allocation for 360° Video (VEGA-360), a hierarchical viewport-aware resource allocation framework for multi-user 360° video streaming. VEGA-360 adopts a two-stage design. On the main stage, VEGA-360 partitions users into a small number of clusters using a joint criterion that combines spectral efficiency and viewport similarity derived from viewport weights and allocates a per-cluster resource budget accordingly. In the fine-tuning phase, VEGA-360 solves independent optimization sub-problems per cluster at the tile granularity, enforcing radio resource and SHVC scalability constraints while maximizing a utility metric that accounts for viewport-weighted visual quality and the transmission overhead caused by distributing multiple tile instances. By separating coarse-grained grouping and budgeting decisions from fine-grained tile-layer allocation, VEGA-360 reduces the size of each optimization instance and improves computational tractability while maintaining viewport-aware service in dense multi-user scenarios. Simulation results show that VEGA-360 achieves competitive utility/QoE compared to a monolithic MILP baseline, with substantially shorter solution times.

KEYWORDS

VEGA-360, QoE, 360-degree video, SHVC encoding, Virtual reality.

1. INTRODUCTION

Virtual Reality (VR) and 360 -degree video technologies have enabled users to experience highly immersive three-dimensional (3D) environments, where they can freely explore and interact with virtual or captured real-world scenes as if they were physically present [1]. By combining head-mounted displays (HMDs), motion tracking and interactive content, VR has been successfully adopted in education, healthcare, manufacturing, entertainment and many other domains, where immersive visual experiences can enhance engagement, training effectiveness and decision-making.

To deliver such immersive experiences at scale, 360 -degree videos are typically captured and rendered in high resolutions (e.g. 4 K and beyond), which leads to extremely large data volumes and high bitrate demands [2]. These requirements become even more stringent on mobile devices, where limited computation capability, constrained battery and fluctuating wireless bandwidth can easily cause playback stalls, blurry viewports or noticeable latency. To cope with these constraints, tile-based viewport-adaptive streaming has emerged as a widely adopted approach: the spherical video is partitioned into multiple tiles and only the tiles within the user's current Field of View (FoV) or Region of Interest (RoI) are delivered at high quality, while the remaining tiles are sent at reduced quality [2]-[3].

Scalable video-coding extensions, such as Scalable High-efficiency Video Coding (SHVC), further enable flexibility by encoding each tile into a base layer and multiple enhancement layers [4]-[5]. The base layer guarantees minimum decodable quality, whereas enhancement layers can be selectively transmitted to refine spatial resolution or improve quality when network conditions permit. At the same time, the evolution of mobile networks from 4G LTE to 5G brings significantly higher peak data rates, lower latency and improved spectral efficiency, offering an attractive infrastructure for delivering interactive 360 -degree video in real time [6]-[7]. As illustrated in Fig. 1, each selected tile is encoded into multiple quality layers and broadcast to different user clusters. High-capability users subscribe to

P. D. Huynh and N. V. Hung (Corresponding author) are with the East Asia University of Technology, Vietnam. Emails: 20222072@eaut.edu.vn and hungnv@eaut.edu.vn

more enhancement layers, while low-capability users only receive the base layer. This layered multi-cast structure serves as the basis for the proposed VEGA-360 optimization framework.

Despite these advances, providing high-quality 360 -degree video streaming to multiple mobile users remains challenging. Existing solutions often optimize either viewport-adaptive tiling [2]-[3],[8] or streaming strategies (uni-cast/multi-cast, rate adaptation) [9][10][11][12], but still face difficulties in simultaneously handling: Firstly, heterogeneous link conditions across users; secondly, diverse viewport dynamics and RoI preferences; and finally, limited radio resources on a single cell. Recent studies have explored RoI-based viewport prediction [13]-[14], clustering users with similar viewing patterns [15] and optimized tile-quality selection for multi-user scenarios [16]. However, there is still a lack of a unified framework that jointly exploits user clustering, scalable tiling and resource allocation to maximize Quality of Experience (QoE) under realistic bandwidth constraints.

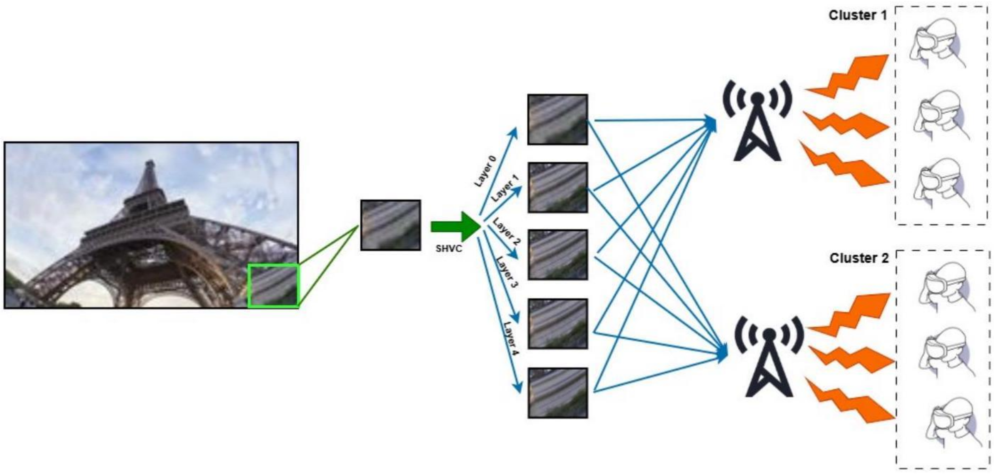

Figure 1. SHVC-based layered multi-cast for tiled 360° video.

Motivated by these limitations, we present VEGA-360, a clustering-based optimization framework for multi-user 360° video streaming over mobile networks. VEGA-360 groups users according to their tile-level quality requirements and channel conditions and allocates radio resources across clusters and tile layers to prioritize high-impact tiles within users' regions of interest (RoIs) while respecting the global bandwidth budget. By jointly leveraging scalable video encoding and intelligent user clustering, VEGA360 effectively balances fairness and efficiency, reduces redundant transmissions and improves overall QoE compared with existing baselines.

The following part provides an expanded overview of the key contributions presented in this paper, highlighting the main ideas, methodological advances and practical implications derived from the study.

- We design a clustering-based system model for multi-user 360 -degree video streaming, where users sharing similar viewport and quality demands are grouped into clusters and tiles are encoded using scalable video coding to support flexible per-tile quality selection.
- We formulate a joint optimization problem that captures tile-layer selection and radio-resource allocation across clusters under bandwidth and QoE constraints, explicitly focusing on tiles lying in users' RoI.
- We develop a practical-solution algorithm to solve the optimization problem segment-by-segment, enabling the system to adapt to time-varying network conditions and dynamic viewports while maintaining stable QoE.
- We conduct extensive simulations using real 360 -degree video traces and head-movement datasets and compare VEGA-360 against other state-of-the-art schemes. The results show that our framework improves QoE and viewport quality while keeping bandwidth usage within practical limits.

This is how the rest of the paper is structured. In Section 2, relevant research on user-clustering techniques and 360 -degree video streaming is reviewed. The system model and problem formulation are presented in Section 3. The suggested framework and solution algorithm are presented in Section 4. Section 5 presents the results of the performance evaluation and Section 6 wraps up the work.

2. RELATED WORK

The rapid growth of immersive services has driven extensive research on adaptive 360 -degree video streaming over wireless networks. Recent work has focused on designing smarter adaptation strategies that consider both network dynamics and user experience. Chen et al. studied streaming 360° VR video with statistical QoS provisioning in mmWave networks, highlighting the role of wireless reliability constraints in immersive delivery [17]. Badnava et al. formulated multi-user 360 -degree video delivery as a multi-task decision-making problem and used a deep reinforcement-learning agent to jointly allocate bitrate and computation resources, with the goal of maximizing long-term QoE under fluctuating bandwidth [18]. Wang et al. adopted a multi-agent deep reinforcement learning framework to control rate adaptation for 360-degree contents, where multiple viewpoints and fairness among users are explicitly modelled in the optimization process [19]. In parallel, Nguyen et al. addressed robustness to sudden throughput reductions and proposed a scalable and resilient 360-degree HTTP/2 streaming solution that exploits stream prioritization and independence to mitigate stalls and quality drops [20]. Other recent studies explored joint optimization of streaming and enhancement. For example, Guo et al. investigated coordinated control of coding parameters and super-resolution filters for mobile 360 -degree delivery [21], while Feng et al. designed a stochastic multi-window adaptation scheme that couples viewport prediction and bitrate assignment [22]. These approaches, however, operate mostly on a per-user basis and do not fully exploit the potential of multi-cast gains when users share similar viewports and quality requirements. FoV overlap has also been explicitly exploited in wireless VR delivery in order to improve robustness and efficiency when users share similar viewing regions [23]. Relatedly, Abedini and Nickray employed reinforcement learning to tune transport-layer congestion control for real-time delivery, indicating that cross-layer adaptation can help stabilize end-to-end performance under time-varying bandwidth [24].

Viewport prediction and user-behavior modeling form another active line of research for 360 -degree video. Wahba et al. provided a recent survey of learning-based viewport-prediction techniques and identified open problems related to latency, generalization and device heterogeneity in practical systems [25]. Building on data-driven prediction, Wang et al. introduced an edge-assisted clustered-learning framework, CoLive, that groups users based on their viewing behaviour and trains cluster-specific models to improve prediction accuracy and streaming efficiency in live scenarios [26]. Zhang et al. proposed a mobile-friendly viewport prediction method for live 360 -degree streaming in which attention-aware features and device constraints are jointly exploited [27]. Besides, Nguyen et al. developed a GRU-LSTM-based viewport-estimation method tailored to 360 -degree video streaming [28]-[29] and more recent work explores reinforcement-learning based viewport estimation that fuses head and eye-movement information (HEVERL) for VR applications [30]. These works clearly show that exploiting correlations between users and leveraging clustering at the prediction layer can improve performance, but they stop short of incorporating clustering directly into a global multi-cast resource-allocation problem.

User clustering for multi-user 360° streaming can be broadly categorized into channel-based and behavior-based clustering. Channel-based clustering groups users according to wireless channel conditions (e.g. spectral efficiency or SNR), which is attractive for multi-cast, because the cluster transmission rate is typically constrained by the worst-channel user. However, clustering solely by channel may ignore viewport heterogeneity and enlarge the union of requested tiles, reducing multi-cast efficiency. In contrast, behavior-based clustering groups users by viewing behaviour (viewport/FoV similarity or predicted viewport trajectories) to maximize tile reuse and reduce redundant transmissions. Representative behavior-aware works emphasized viewport prediction and viewport-adaptive delivery, including SPA360 [31], Meta360 [32], FoV prediction-assisted viewport delivery [33] and utility-driven optimization in JUST360 [34]. While these studies highlighted the benefits of behavior awareness, they generally did not integrate channel heterogeneity into the clustering decision for multi-cast resource allocation. Our work bridges this gap by explicitly combining channel information and viewport similarity in the clustering stage and coupling them with hierarchical per-cluster resource allocation for multi-cast tiled SHVC delivery.

Accurate QoE modeling for immersive media has also received increased attention. More recently, physiological-signal-driven QoE optimization has been investigated for wireless VR transmission, providing an alternative direction for modeling user-perceived experience beyond traditional quality metrics [35]. Nguyen et al. proposed a retina-inspired objective quality-assessment model for tile-coded 360 -degree videos, in which spatial weights are aligned with the non-uniform sensitivity of the human visual system across the field of view [36]. Elwardy et al. presented a pilot study on the consistency of subjective quality assessment for 360 -degree contents, introducing the RQA360 dataset and analyzing repeated tests in both standing and seated viewing conditions [37]. Complementary evidence is reported by Qananwah et al., who explored physiological cues (EEG signals) to guide video-compression decisions, reinforcing the value of human-centric information when constructing QoE-aware streaming and coding strategies [38]. These studies highlighted the importance of QoE metrics that not only reflect the delivered quality level, but also account for viewport importance, spatial-quality variations across tiles and the impact of experimental settings such as viewing posture and device. Such insights motivate the use of viewport-weighted utility functions and penalty terms in the design of optimization-based streaming frameworks. Closer to the present work, clustering-based optimization for 360-degree multi-cast has been investigated from a cross-layer perspective. Nguyen et al. proposed a clustering-based framework for scalable multi-cast of tiled 360-degree videos in multi-cell wireless networks, in which a mixed-integer linear program jointly selects SHVC layers and user clusters under resource constraints [39]. That monolithic approach demonstrates notable QoE and bandwidth gains compared with uni-cast and heuristic baselines, but its computational complexity grows rapidly with the number of users, tiles and available resource blocks, which limits scalability in dense deployments. In contrast, the VEGA-360 framework studied in this paper adopts a hierarchical two-stage design: a first-stage viewport- and channel-aware clustering step groups users and allocates a per-cluster resource budget based on users' spectral efficiencies and viewport similarity (derived from tile weights) and independent second-stage sub-problems allocate tile versions and radio resources within each cluster.

3. System Model and Design Overview

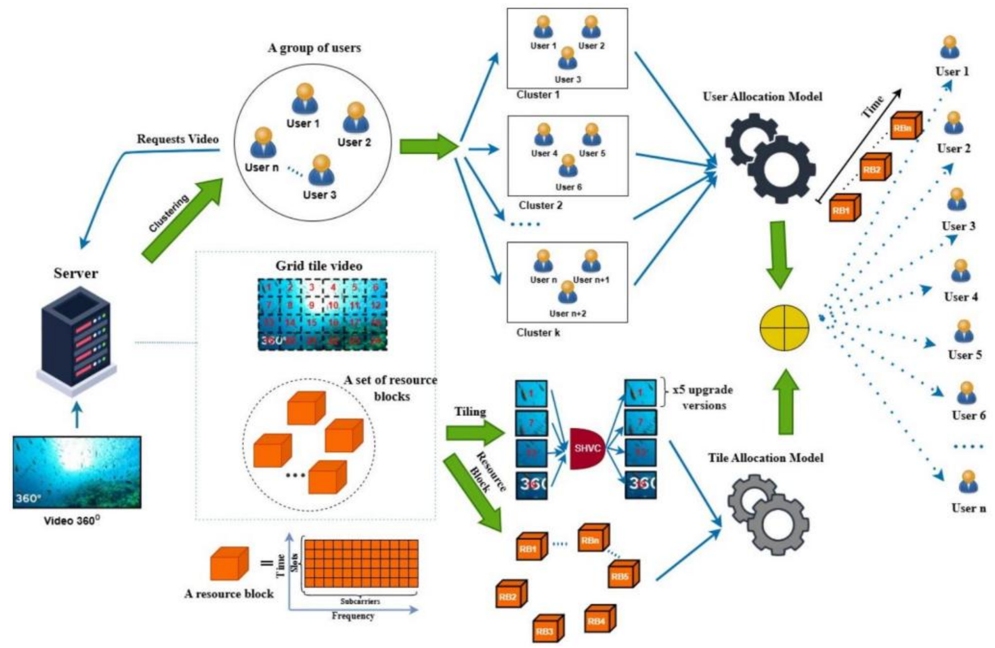

Figure 2. Overall architecture of the proposed VEGA-360 system. The server performs tiling and SHVC encoding, the base station groups users into clusters according to channel quality and viewport similarity and a joint user/tile allocation model maps SHVC layers to resource blocks.

The system model of the suggested VEGA-360 framework for QoE-aware 360-degree video multi-cast, as seen in Fig. 2 and Fig. 3, is described in this section.

We consider a wireless downlink scenario in which a video server stores several 360° videos and communicates with a group of heterogeneous users through a base station (BS). All users request the same 360° video segment at a given time. The processing begins by dividing the video into spatial tiles and temporal segments. Let

$$\mathcal{U} = \{1, \dots, U\}, \mathcal{T} = \{1, \dots, T\} \tag{1}$$

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 12, No. 01, March 2026.

represent the sets of users and tiles within a single segment, respectively. Each tile is encoded by SHVC into C scalable layers, indexed by $c \in \{0, \dots, C-1\}$, where $c = 0$ is the base layer and $c \geq 1$ are enhancement layers. The c-th layer provides an objective video quality Q_c (e.g. PSNR) and requires bitrate (or bandwidth) B_c.

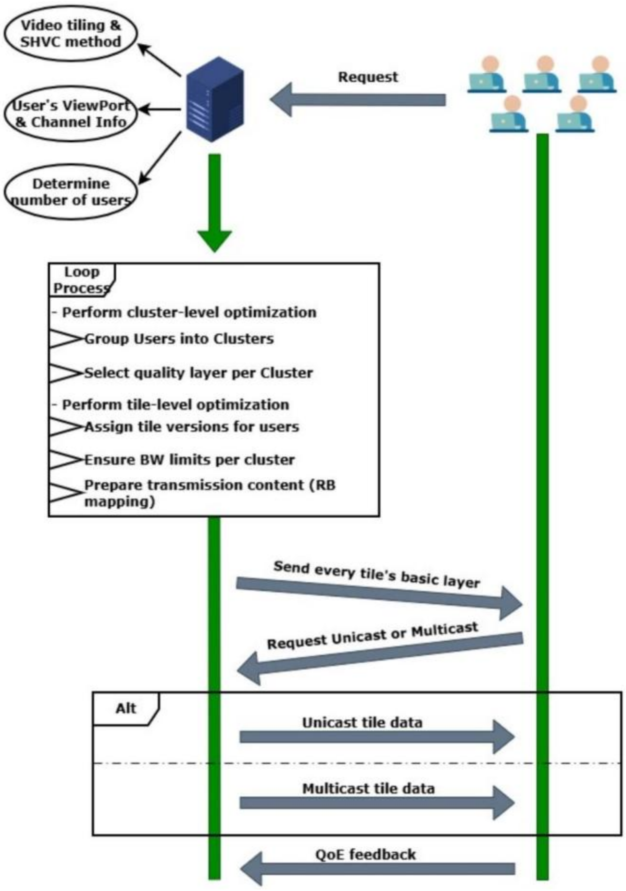

Figure 3. Sequence diagram of the VEGA-360 streaming workflow between server and client sides for one video segment.

For each user $u \in \mathcal{U}$, we model the importance of tile $t \in \mathcal{T}$ by a weight $w_{u,t} \in [0,1]$, which reflects how likely tile t lies inside the viewport of user u. The weights are normalized as:

$$\sum_{t \in \mathcal{T}} w_{u,t} = 1, \forall u \in \mathcal{U}. \tag{2}$$

Here, large $w_{u,t}$ means that tile t contributes more to the perceived QoE of user u. The wireless channel of user u is characterized by an average spectral efficiency σ_u (bit/s/Hz), which captures path loss, fading and the selected modulation and coding scheme.

The BS is allocated a total downlink resource budget R per segment, measured, for instance, in OFDM resource blocks (RBs). As depicted in Fig. 2, each RB occupies a certain time-frequency area and can carry coded bits of one or several tile layers. The overall resource constraint is expressed as:

$$\sum_{k=1}^{K} \sum_{c=0}^{C-1} R_{k,c} \leq R, \tag{3}$$

where K is the number of user clusters and $R_{k,c}$ denotes the number of RBs used to transmit the c-th layer that is multi-cast to cluster k. This constraint couples the decisions of quality selection and user grouping across all clusters.

To exploit multi-cast gain while preserving individual QoE, VEGA-360 partitions users into K clusters based on a joint criterion that combines spectral efficiency and viewport similarity derived from $\{w_{u,t}\}$.

$$\mathcal{K} = \{1, \ldots, K\}, \tag{4}$$

Users in the same cluster are served by a common multi-cast stream and thus tend to receive similar quality layers, whereas different clusters may receive different numbers of SHVC layers depending on their channel conditions and the global budget R. Clustering reduces the signaling overhead and makes the subsequent optimization scalable when the number of users grows.

The end-to-end operation for one video segment is summarized in Fig. 3. First, the users send a request for a $360°$ video and the server performs tiling and SHVC encoding of the current segment. Second, the BS collects viewport statistics (to update the weights $w_{u,t}$) and channel information (to update the spectral efficiencies σ_u) and determines the current number of active users U. Based on this information, the BS groups users into clusters and allocates a per-cluster resource budget R_k under the total budget R. Next, for every cluster, the BS refines the decision at tile level by assigning, for each tile t and user u in that cluster, which tile version (i.e., which layer index c) should be transmitted, so that all users decode at least the base layer and the per-cluster bandwidth limits are satisfied. Afterwards, the BS prepares the actual transmission content by mapping the selected tile layers onto RBs and delivers them over the air interface. When the same coded tile layer is requested by multiple users in a cluster, it is sent *via* multi-cast; otherwise, individual tiles can be complemented *via* uni-cast when necessary. Finally, after playback, the system may collect QoE-related feedback from users, which can be exploited for long-term adaptation of clustering and resource allocation.

Algorithm 1 summarizes the main-stage clustering of VEGA-360. Unlike purely channel-based grouping, VEGA-360 jointly considers users' spectral efficiencies and viewport similarity derived from the tile-weight vectors $\mathrm{w}_u = \left[w_{u,0}, \ldots, w_{u,T-1}\right]$. This design prevents grouping users who have similar channel conditions, but request disjoint viewports, which would otherwise reduce multi-cast gain and waste bandwidth due to union-tile transmissions. After forming clusters, VEGA-360 allocates a per-cluster budget R_k and then performs tile-level optimization within each cluster.

Algorithm 1 Viewport and channel-aware clustering in VEGA-360

1: Collect users' viewport weights $\{w_{u,t}\}$ and spectral efficiencies $\{\sigma_u\}$.
2: Initialize K clusters $\{U_k\}_{k=0}^{k-1}\}$ with capacities $\{L_k\}$.
3: Initialize each cluster's centroid viewport vector $\overline{w_k}$ and average channel $\overline{\sigma_k}$ (e.g., using seed users).
4: **for** each user u (in descending order of σ_u or in any fixed order) do
5: **for** each cluster k with $|U_k| < L_k$ **do**
6: Compute viewport similarity $s_{u,k}^{vp} \leftarrow \mathrm{sim}(w_u, \overline{w_k})$.
7: Compute channel similarity $s_{u,k}^{ch} \leftarrow |\sigma_u - \overline{\sigma_k}|$.
8: Compute joint score $S_{u,k} \leftarrow \alpha s_{u,k}^{ch} + (1-\alpha)\, s_{u,k}^{vp}$.
9: **end for**
10: Assign user u to the cluster k^* = $\mathrm{argmax}_k S_{u,k}$.
11: **Update** $\overline{w_{k^*}}$ and $\overline{\sigma_{k^*}}$.
12: **end for**
13: Allocate per-cluster budgets $\{R_k\}$ under the total budget R.
14: **return** $\{U_k\}$ and $\{R_k\}$.

4. PROPOSED VEGA-360 METHOD

4.1 Overview of Tile Transportation

In this sub-section, we present the proposed VEGA-360 framework for QoE-driven multi-cast and uni-cast delivery of tiled $360°$ video. We first use Fig. 4 to explain how VEGA- 360 jointly handles SHVC layers, tiles and time segments. Then, we formulate the main optimization problem and highlight the key constraints that govern user clustering, layer selection and tile-level allocation.

Fig. 4 illustrates the scheduling structure of VEGA-360. The original $360°$ frame is divided into a grid of tiles, indexed by $t \in \mathcal{T} = \{1, \ldots, T\}$. Each tile is encoded into C SHVC layers, denoted by $\{v_0, \ldots, v_{C-1}\}$, where v_0 is the base layer and higher indices correspond to higher visual quality and higher bitrate.

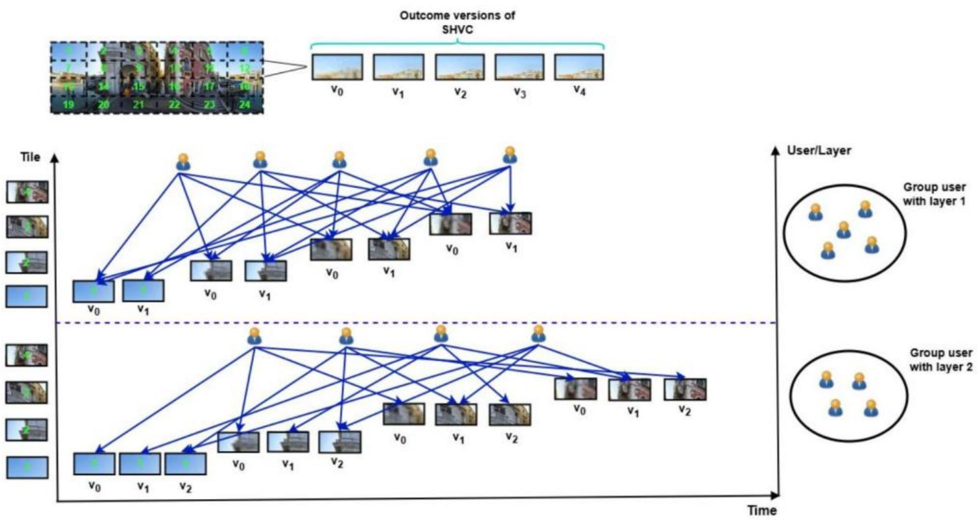

Figure 4. The VEGA-360 structure illustrated with user clusters on the right, time segments on the horizontal and tiles on the vertical.

On the user side, we denote the user set by $\mathcal{U} = \{1, \ldots, U\}$. For each user $u \in \mathcal{U}$ and each tile $t \in \mathcal{T}$, we pre-compute a weight $w_{u,t}$ that reflects how likely tile t falls into the user's viewport. Tiles that frequently appear in the viewport are assigned larger weights, while tiles rarely watched by user u have very small weights. Along the horizontal axis in Fig. 4, time is divided into segments (chunks) of equal duration. For every segment, VEGA-360 first groups users into clusters and allocates a per-cluster resource budget R_k and then decides which SHVC layer is transmitted for each user-tile pair within each cluster.

Users with similar channel conditions and viewport characteristics are dynamically grouped into clusters. In Fig. 4, the right-hand side shows two such clusters. Inside each cluster, the base layer v_0 of the required tiles is always transmitted to guarantee decodability, while enhancement layers $v_1, \ldots, v_{C-1}$ are selectively delivered depending on viewport weights and channel conditions, as depicted by the blue arrows.

4.2 Main Formulae Contribution

For each user $u \in \mathcal{U}_k$, tile $t \in \{0, \ldots, T-1\}$ and version index $c \in \{0, \ldots, C_k - 1\}$, we introduce a binary decision variable

$$y_{u,t,c} = \begin{cases} 1, \text{ if version } c \text{ of tile } t \text{ is delivered to user } u, \\ 0, \text{ otherwise .} \end{cases} \tag{5}$$

The per-cluster viewport quality for one segment is computed as:

$$VQ_k = \sum_{u \in \mathcal{U}_k} \sum_{t=0}^{T-1} \sum_{c=0}^{C_k - 1} w_{u,t} Q_c y_{u,t,c}, \tag{6}$$

where VQ_k measures the accumulated quality perceived by all users in cluster $k, w_{u,t}$ is the normalized importance of tile t in the viewport of user u and Q_c denotes the objective quality (e.g. PSNR) of version c.

In parallel, we keep track of how many tile versions are transmitted in cluster :

$$TV_k = \sum_{u \in \mathcal{U}_k} \sum_{t=0}^{T-1} \sum_{c=0}^{C_k - 1} y_{u,t,c}, \tag{7}$$

where TV_k acts as a proxy for transmission overhead and decoding complexity, since each selected version corresponds to an additional bitstream that must be sent and decoded.

The global QoE metric of VEGA-360 combines viewport quality and transmission cost as:

$$d_k \triangleq \frac{\tau_{\text{seg}}}{R_k} \sum_{u \in U_k} \sum_{t} \sum_{c} \frac{\left(y_{u,t,c} - z_{u,t,c}\right) B_c}{\sigma_u} \tag{8}$$

$$r_u \geq d_{k(u)} - b_u, r_u \geq 0 \tag{9}$$

$$\text{QoE} = \frac{\alpha}{1000} \sum_{k=0}^{K-1} VQ_k - \frac{\gamma}{1000} \sum_{k=0}^{K-1} TV_k - \frac{\beta}{1000} \sum_{u=1}^{U} r_u \tag{10}$$

where $\alpha > 0$ weights viewport quality and $\gamma > 0$ penalizes transmission cost (number of delivered tile versions). d_k is the effective delivery time of cluster k for one segment and r_u denotes the rebuffering time of user u, modelled by the linear constraints in Eq. (9) (i.e., $r_u = \left[d_{k(u)} - b_u\right]^+$). $\beta > 0$ weights the stalling penalty. We set $\alpha = 1, \gamma = 1$ and $\beta = 1.85$ and apply a normalization factor of $1/1000$ for numerical stability.

Each cluster is subject to its own radio-budget constraint. The resource consumption of cluster k is upper bounded by R_k:

$$\sum_{u \in \mathcal{U}_k} \sum_{t=0}^{T-1} \sum_{c=0}^{C_k-1} \frac{B_c}{\sigma_u} y_{u,t,c} \leq R_k, \tag{11}$$

where B_c is the bitrate of version c, σ_u denotes the spectral efficiency of user u and R_k is the share of available radio resources allocated to cluster k. This constraint guarantees that the cumulative bandwidth of the selected tile versions does not surpass the cluster's allocated budget.

In our implementation, the per-cluster budget R_k is obtained by splitting the total budget R proportionally to the minimum base-layer delivery cost of each cluster. Specifically, we compute

$$W_k = \sum_{u \in U_k} \frac{B_0}{\sigma_u}, R_k = R \cdot \frac{W_k}{\sum_{j=0}^{K-1} W_j}, \tag{12}$$

where B_0 is the bitrate of the base layer. This allocation assigns more resources to clusters with poorer channel conditions (smaller σ_u), ensuring base-layer feasibility and avoiding starvation of low-capability users before the tile-level optimization in Eq. (16).

Due to the hierarchical structure of SHVC, an enhancement version can only be decoded if all lower versions of the same tile are also available. VEGA-360 enforces this dependency through

$$y_{u,t,c} \leq y_{u,t,c-1}, \forall u \in \mathcal{U}_k, \forall t, c = 1, \ldots, C_k - 1, \tag{13}$$

which ensures that whenever version c is selected, version $c - 1$ is selected as well. Moreover, the basic visibility of the $360°$ scene is always guaranteed by forcing the base layer of every tile to be sent:

$$y_{u,t,0} = 1, \forall u \in \mathcal{U}_k, \forall t. \tag{14}$$

Finally, all decision variables are binary,

$$y_{u,t,c} \in \{0,1\}, \forall u \in \mathcal{U}_k, \forall t, \forall c, \tag{15}$$

so that each version of each tile is either fully selected or not transmitted. Putting everything together, the tile-level optimization in cluster k is written as

$$\max_{\{y_{u,t,c}\}} \alpha VQ_k - \gamma TV_k \text{ s.t. Eq.11-15}, \tag{16}$$

and is solved for every cluster under its corresponding budget R_k. To further exploit the heterogeneity of viewports inside a cluster, VEGA-360 refines the solution of Eq. (16) by imposing a QoE-aware ordering across users, as summarized in Algorithm 2. For each tile t, the users in $\mathcal{U}_k$ are first sorted in descending order of their viewport weights $w_{u,t}$, obtaining a sequence $\left(u_0, u_1, \ldots, u_{|\mathcal{U}_k|-1}\right)$ from the most to the least-interested user. Then, for every adjacent pair (u_i, u_{i+1}) and every admissible version c, the inequality

$$y_{u_{i+1},t,c} - y_{u_i,t,c} \leq 0$$

is enforced. This simple rule guarantees that a user with lower importance for tile t never receives a higher version than a user with higher importance in the same cluster. As a result, VEGA-360 creates a staircase pattern of tile versions inside each cluster: central viewports are upgraded first, while the multi-cast structure is preserved, leading to a better QoE-bandwidth trade-off.

Algorithm 2 QoE-aware ordering of tile versions in cluster k

for t = 0 to T − 1 **do**
  Sort users in u_k by descending $w_{u,t}$ obtain the sequence $(u_1, u_2, \ldots, u_{|u_k|-1})$
  for i = 0 to $|u_k| - 1$ **do**
    if i + 1 < $|u_k|$ **then**
      for c = 0 to $C_k - 1$ **do**
        $y_{u_{i+1},t,c} - y_{u_i,t,c} \le 0$
      end for
    end if
  end for
end for

5. PERFORMANCE EVALUATION

In this section, we first describe the 360 -degree video dataset, encoding configuration and simulation parameters used to evaluate VEGA-360 and then discuss the obtained performance in comparison with existing baselines.

5.1 Experimental Setup

Our performance evaluation is carried out in a custom simulator implemented in Python, where we replay the clustering and scheduling decisions of VEGA-360 together with the baseline algorithms. All mixed integer programs are solved by the Gurobi optimizer on a standard workstation equipped with an Intel Core i7 CPU and 32 GB of RAM, with a moderate time limit per instance.

We reuse a public 360 -degree video dataset introduced in [40], which contains five omni-directional sequences: Rollercoaster, Diving, Venice, Paris and Rhino. Following the original dataset, the videos are grouped into two content categories: "less-feature" (mostly static scenes) and "more-feature" (dynamic scenes with many moving objects and camera motion). Each raw video is projected to the equirectangular format with a resolution of 2890×1920 pixels and partitioned into $T = 24$ tiles of size 480×480 pixels per tile. To support scalable streaming, every tile is encoded with the SHVC extension of HEVC into one base layer and four enhancement layers, i.e., $C = 5$ quality versions per tile. Table 1 summarizes the average viewport PSNR (in dB) and bitrate (in kbps) of the five versions for the two content categories, averaged over all videos in the dataset.

We consider a single cell serving $U = 70$ mobile users requesting the same 360 -degree video. User-specific viewport trajectories are obtained from the head-movement traces. For each segment, we project the viewport field-of-view onto the tiled equirectangular plane and compute tile weights by the normalized viewport-tile overlap ratios, so that the weights reflect how much each user watches each tile in that segment. Following the COSMN setting, we generate $U = 70$ users by sampling traces with replacement and randomizing the starting segment index to avoid synchronized viewing patterns. The wireless downlink is abstracted by a total resource budget per segment shared by all users and clusters and the budget is swept from 10000 to 120000 (arbitrary resource units) to emulate different congestion levels. Users' spectral efficiencies follow the same channel abstraction as COSMN [39] and are used to determine per-user transmission cost in the optimization. For the rebuffering term, we use a fixed segment duration and a fixed initial playback buffer as Eq. (17):

$$\tau_{\text{seg}} = 1 \text{ s}, b_u = b_0 = 2 \text{ s}, \forall u \in \{1, \ldots, U\}. \quad (17)$$

For each value of R, we run VEGA-360 and four baseline schemes under the same $w_{u,t}$ and encoding parameters:

- COSMN [39]: the original clustering-based optimization that solves a single global MILP over all users, tiles and versions.

- LVSUM [9]: a greedy layer-by-layer strategy that upgrades tile versions sequentially according to the remaining budget.
- Multi-cast All [41]: a multi-cast-only scheme that delivers the same version of each tile to every user, without exploiting viewport diversity.
- Multi-cast Sca [10]: a scalable multi-cast scheme that uses the SHVC layers, but still ignores fine-grained viewport heterogeneity.

Table 1. Average viewport PSNR and bitrate of the encoded tile versions.

Values	Ver. 0	Ver. 1	Ver. 2	Ver. 3	Ver. 4
Less-feature videos					
PSNR (dB)	36.79	41.09	42.89	45.64	48.30
Bitrate (kbps)	58.57	148.37	281.69	560.84	995.61
More-feature videos					
PSNR (dB)	35.30	38.70	41.45	44.71	47.34
Bitrate (kbps)	177.40	417.32	741.89	1357.76	2143.49

All schemes are evaluated using the QoE metric defined in Eq. (10), which linearly trades-off viewport quality against the number of transmitted tile versions. In addition, we also report the average viewport PSNR (VQ) and the average number of transmitted tile versions per segment (TV) as auxiliary indicators.

5.2 Results and Discussion

On the one hand, Table 2 displays the average quality of experience measured for the video with fewer features. VEGA-360 achieves consistently strong QoE across feasible bandwidth budgets and it remains competitive (often best) when the bandwidth budget increases. This behavior indicates that VEGA-360 can sustain user-perceived quality even when user population grows, while still respecting the available radio resources. In addition, the performance gap among optimization-based methods becomes small at medium-to-high budgets, suggesting that QoE gradually saturates once most viewport-important tiles can be delivered at adequate quality.

Table 2. Average QoE for the "less-feature" video under different bandwidth budgets and numbers of users.

Method	Users	Bandwidth budget (kRBs)						
		20	30	45	55	65	75	80
VEGA-360	35	-	-	1.949	1.955	1.969	1.966	1.971
	70	-	-	2.610	2.614	2.632	2.625	2.637
COSMN	35	1.924	1.936	1.947	1.954	1.960	1.965	1.967
	70	2.584	2.596	2.607	2.613	2.620	2.624	2.626
LVSUM	35	1.917	1.927	1.943	1.949	1.957	1.962	1.964
	70	2.577	2.586	2.602	2.609	2.616	2.621	2.623
Multicast All	35	-	-	-	-	-	1.873	1.880
	70	-	-	-	-	-	2.532	2.540
Multicast Sca	35	-	-	-	-	-	1.880	1.891
	70	-	-	-	-	-	2.540	2.550

On the other hand, Table 3 summarizes the results for the more-feature video. Compared to the less-feature case, the more-feature content typically requires higher delivery effort to maintain the same perceived quality, which makes the bandwidth budget more influential. VEGA-360 remains robust across a broader range of budgets and user scales, demonstrating stable QoE when scaling from 35 users to 70 users.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 12, No. 01, March 2026.

Overall, the results highlight that the proposed design maintains competitive QoE under heterogeneous user demands and richer visual details.

To better illustrate the resource aspect, Fig. 5 and Fig. 6 visualize the bandwidth consumption under different user populations. In general, as the number of users increases, purely unicast-oriented strategies tend to incur higher bandwidth usage. By contrast, VEGA-360 is designed to exploit multi-cast opportunities through QoE-aware ordering and reuse of tile versions, thereby limiting redundant transmissions while preserving viewport quality. This multicast-aware behavior is more evident in the more-feature setting, where the content complexity and quality requirements amplify the benefit of coordinated version delivery.

Finally, Fig. 8 and Fig. 7 present the PSNR comparison across bandwidth budgets for each method. Overall, VEGA-360 achieves competitive (and often higher) reconstruction quality, especially in the low-to-medium budget region where bandwidth scarcity makes tile/version prioritization critical. As the budget increases, the PSNR gap among optimization-based schemes becomes smaller, indicating a saturation effect: once most viewport-relevant tiles can be delivered at sufficiently high quality, additional bandwidth yields marginal visual gains. These results support the main goal of VEGA-360, i.e., improving visual fidelity (PSNR) while avoiding unnecessary bandwidth inflation in 360° tiled streaming.

Table 3. Average QoE for the "more-feature" video under different bandwidth budgets and numbers of users.

Method	Users	Bandwidth budget (kRBs)						
		50	80	120	180	250	280	300
VEGA-360	35	-	1.901	1.905	1.922	1.929	1.935	1.932
	70	-	2.516	2.541	2.547	2.570	2.576	2.576
COSMN	35	1.870	1.889	1.902	1.919	1.929	1.931	1.931
	70	2.515	2.533	2.547	2.564	2.573	2.575	2.576
LVSUM	35	1.858	1.879	1.896	1.915	1.926	1.929	1.930
	70	2.503	2.524	2.541	2.560	2.571	2.574	2.575
Multicast All	35	-	-	-	-	1.828	1.835	1.840
	70	-	-	-	-	2.473	2.480	2.484
Multicast Sca	35	-	-	-	-	1.838	1.849	1.855
	70	-	-	-	-	2.483	2.494	2.500

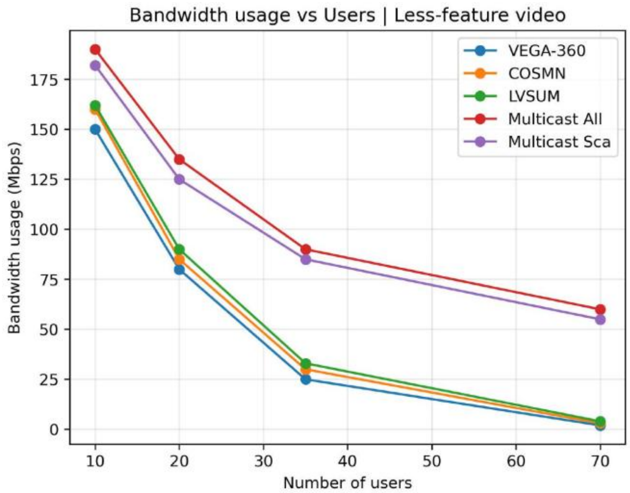

Figure 5. Bandwidth consumption *versus* number of users for the "less-feature" video. VEGA-360 consistently requires lower bandwidth than COSMN and other baselines under the same user load.

We also present Table 4 that reports the average solver runtime of our method and the baselines under different total radio budgets R. Overall, our method consistently achieves lower runtime than COSMN across all tested R values (bold entries). In particular, while COSMN requires about $0.55 - 1.00$ second, our method finishes within $0.28 - 0.55$ second, showing a clear reduction in decision latency as the budget increases. The runtimes of the other baselines (LMSUM, Mul_all and Mul_sca) lie between COSMN and our method in most cases.

After analyzing the outcomes, we identify several practical remarks emerging from the experimental evaluation:

- First, the benefits of VEGA-360 are most evident in the low-to-medium bandwidth region, where resource scarcity forces strict prioritization between tiles and instances; in such regimes, coordinated instance reuse and multi-cast distribution avoid redundant transmissions while maintaining image quality.
- Second, the PSNR curves tend to saturate as bandwidth budgets increase, which suggests diminishing returns once the majority of view-relevant tiles are already delivered at sufficiently high quality; thus, pursuing aggressive upgrades at high budgets is less beneficial than improving efficiency at tight budgets.
- Third, scaling the number of users amplifies the advantages of multi-directional awareness designs: while unidirectional-focused strategies will generate bandwidth growth that is roughly proportional to the number of users, VEGA-360 can limit additional bandwidth by reusing cell instances within groups whenever users share similar viewing preferences.

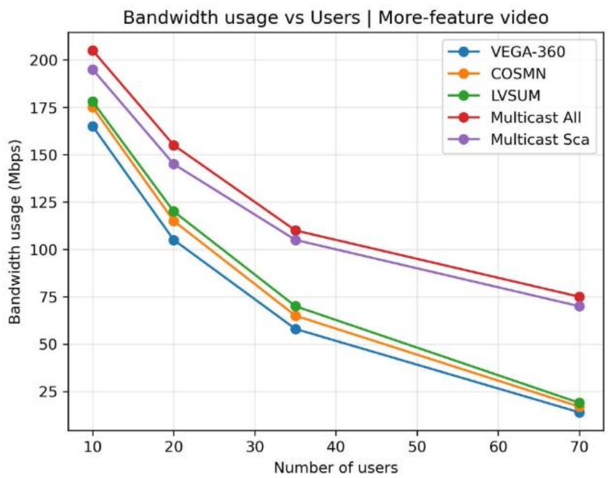

Figure 6. Bandwidth consumption *versus* number of users for the "more-feature" video. The proposed VEGA-360 maintains the best bandwidth efficiency across all evaluated user scales.

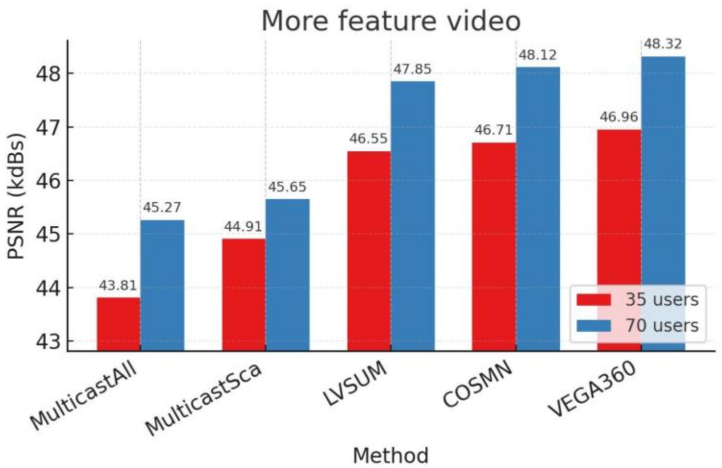

Figure 7. Viewport-quality comparison for the "more-feature" video in terms of PSNR. Results are shown for 35 users and 70 users, highlighting VEGA-360's advantage over COSMN and multi-cast baselines.

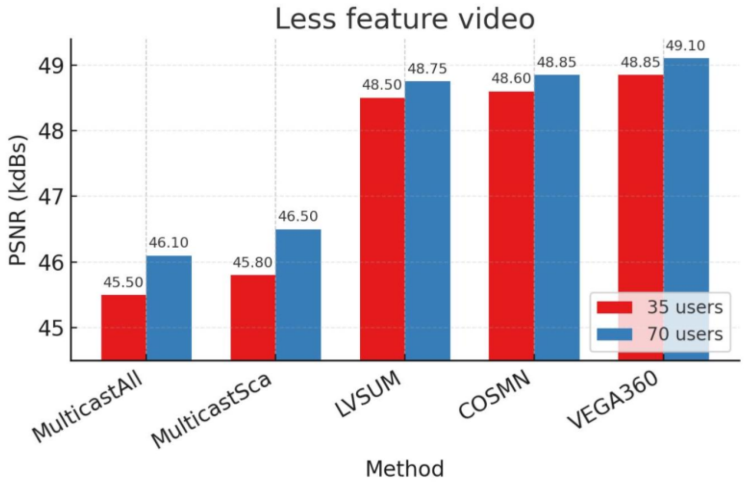

Figure 8. Viewport quality comparison for the "less-feature" video in terms of PSNR. VEGA360 achieves the highest PSNR for both 35- and 70-user scenarios.

- Finally, these observations indicate that VEGA-360 is particularly well-suited to scenarios with multiple users and bandwidth-constrained wireless systems, where improving the quality-efficiency trade-off has a larger impact than marginal quality gains at resfource-rich levels.

Table 4. Runtime comparison (second) under different total radio budgets R.

R	COSMN	LMSUM	Mul_all	Mul_sca	Our method
20000	0.55	0.48	0.30	0.44	0.28
30000	0.62	0.54	0.33	0.48	0.31
45000	0.71	0.61	0.47	0.53	0.36
55000	0.82	0.70	0.51	0.57	0.41
65000	0.90	0.78	0.64	0.68	0.46
75000	0.96	0.83	0.66	0.73	0.52
80000	1.00	0.86	0.78	0.85	0.55

6. CONCLUSION

In this paper, we present VEGA-360, a QoE-aware delivery framework for tile-based $360°$ video streaming over bandwidth-constrained wireless networks. VEGA-360 is designed to balance image quality and transmission performance by coordinating tile-based version selection and leveraging multi-cast opportunities across user groups. Experimental results on representative $360°$ content across different bandwidth budgets and user sizes show that VEGA-360 achieves competitive performance compared to baseline schemes, especially in bandwidth-constrained regimes where efficient reuse of delivered versions is crucial. As a two-stage framework, VEGA-360 trades global optimality for computational tractability and the clustering decision may introduce a small performance gap compared to a monolithic formulation in certain cases. In addition, our current setting assumes a shared-content scenario where users request the same $360°$ video; multi-cast gains may decrease in heterogeneous on-demand scenarios.

In the future, we plan to extend VEGA-360 in three directions: first, integrating online-view prediction and prediction-fault tolerance; next, exploring adaptive clustering and dynamic update intervals under rapidly changing channel and mobile user conditions; and finally, deploying a real-time prototype to evaluate end-to-end delay and system overhead in real networks. We will also investigate more flexible grouping mechanisms (e.g. adaptive or soft cluster-size constraints) to further improve robustness and efficiency.

REFERENCES

[1] A. J. Nair et al., "Unleashing Digital Frontiers: Bridging Realities of Augmented Reality, Virtual Reality and the Metaverse," The Metaverse Dilemma: Challenges and Opportunities for Business and Society, Emerald Publishing Limited, p. 122024, DOI: 10.1108/978-1-83797-524-220241006, 2024.

[2] J. Tu et al., "Pstile: Perception-sensitivity-based 360° Tiled Video Streaming for Industrial Surveillance," IEEE Transactions on Industrial Informatics, vol. 19, no. 9, pp. 9777-9789, 2023.
[3] K. K. Sreedhar et al., "Viewport-adaptive Encoding and Streaming of 360-degree Video for Virtual Reality Applications," Proc. of the 2016 IEEE (ISM), pp. 583-586, San Jose, USA, 2016.
[4] C.-H. Yeh et al., "Fast Prediction for Quality Scalability of High Efficiency Video Coding Scalable Extension," Journal of Visual Communication and Image Representation, vol. 58, pp. 462-476, 2019.
[5] J. M. Boyce et al., "Overview of SHVC: Scalable Extensions of the High Efficiency Video Coding Standard," IEEE Trans. on Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 20-34, 2015.
[6] M. J. Mohammed et al., "A Comparison of 4G LTE and 5G Network Cybersecurity Performance," Proc. of the 2024 35th Conf. of Open Innovations Association (FRUCT), pp. 452-464, Tampere, Finland, 2024.
[7] L. Sun et al., "Multi-path Multi-tier 360-degree Video Streaming in 5G Networks," Proc. of the 9th ACM Multimedia Systems Conf., pp. 162-173, DOI: 10.1145/3204949.3204978, 2018.
[8] N. Al-Najdawi, "High Performance Block Matching Algorithm for High Bit-rate Real-time Video Communication," Jordanian J. of Computers and Inf. Tech. (JJCIT), vol. 4, no. 1, pp. 10-24, 2018.
[9] N. V. Hung et al., "LVSUM-Optimized Live 360 Degree Video Streaming in Unicast and Multicast over Mobile Networks," Proc. of the 2023 IEEE 15th Int. Conf. on Computational Intelligence and Communication Networks (CICN), pp. 29-34, Bangkok, Thailand, 2023.
[10] D. Nguyen, N. V. Hung, N. T. Phong, T. T. Huong and T. C. Thang, "Scalable Multicast for Live 360-degree Video Streaming over Mobile Networks," IEEE Access, vol. 10, pp. 38802-38 812, 2022.
[11] D. T. Nguyen, T. H. Tran and V. H. Nguyen, "Mellifluous Viewport Bitrate Adaptation for 360° Videos Streaming over HTTP/2," ICT Research, [Online], Available: https: //api.semanticscholar.org/CorpusID: 273740657, 2024.
[12] N. V. Hung, B. D. Tien, T. T. T. Anh, P. N. Nam and T. T. Huong, "An Efficient Approach to Terminate 360-video Stream on HTTP/3," Proc. of AIP Conf. Proc., vol. 2909, no. 1, p. 050004, AIP Publish, 2023.
[13] X. Feng, W. Li and S. Wei, "LiveROI: Region of Interest Analysis for Viewport Prediction in Live Mobile Virtual Reality Streaming," Proc. of the 12th ACM Multimedia Systems Conf., pp. 132-145, DOI: 10.1145/3458305.3463378, 2021.
[14] N. Hung et al., "Building an Online Learning Model through a Dance Recognition Video Based on Deep Learning," Informatics and Automation, vol. 23, no. 1, pp. 101-128, 2024.
[15] M. R. Civanlar, Proc. of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, ser. ACM Conferences, New York, NY: Association for Computing Machinery, 2020.
[16] M. Mahmoud et al., "Optimized Tile Quality Selection in Multi-user 360° Video Streaming," IEEE Open Journal of the Communications Society, vol. 5, pp. 7301-7316, 2024.
[17] Y. Chen et al., "Streaming 360° VR Video with Statistical QoS Provisioning in mmWave Networks from Delay and Rate Perspectives," IEEE Trans. on Wireless Comm., vol. 24, no. 6, pp. 4721-4737, 2025.
[18] B. Badnava, J. Chakareski and M. Hashemi, "Multi-task Decision-making for Multi-user 360 Video Processing over Wireless Networks," Proc. of the 2024 IEEE 7th Int. Conf. on Multimedia Information Processing and Retrieval (MIPR), pp. 294-300, DOI: 10.1109/MIPR62202.2024.00054, 2024.
[19] H. Wang, Z. Long, H. Dong and A. El Saddik, "MADRL-based Rate Adaptation for 360° Video Streaming with Multi-viewpoint Prediction," IEEE IoT J., vol. 11, no. 15, p. 26503-26517, Aug. 2024.
[20] V. H. Nguyen et al., "Scalable and Resilient 360-degree-video Adaptive Streaming over HTTP/2 against Sudden Network Drops," Computer Communications, vol. 216, pp. 1-15, 2024.
[21] H. Guo et al., "Joint Adaptation for Mobile 360-degree Video Streaming and Enhancement," IEEE Transactions on Mobile Computing, vol. 24, no. 8, pp. 7726-7741, 2025.
[22] W. Feng, S. Wang and Y. Dai, "Adaptive 360-degree Streaming: Optimizing with Multi-window and Stochastic Viewport Prediction," IEEE Trans. on Mobile Computing, vol. 24, no. 7, pp. 5903-5915, 2025.
[23] J. Lee, H. Lu and Y. Chen, "Robust Wireless VR Video Transmission Based on Overlapped FoVs," Proc. of the IEEE Int. Conf. on Communications (ICC 2023), pp. 3084-3089, Rome, Italy, 2023.
[24] M. N. Ehsan Abedini, "Cubic-learn: A Reinforcement Learning Approach to Cubic Congestion Control," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 11, no. 4, pp. 466 483, 2025.
[25] M. Z. A. Wahba, S. Baldoni and F. Battisti, "Learning-based Viewport Prediction for 360-degree Videos: A Review," Electronics, vol. 14, no. 18, p. 3743, 2025.
[26] M. Wang et al., "CoLive: Edge-assisted Clustered Learning Framework for Viewport Prediction in 360° Live Streaming," IEEE Transactions on Multimedia, vol. 26, pp. 5078-5091, 2024.
[27] L. Zhang et al., "Optimizing Mobile-friendly Viewport Prediction for Live 360-degree Video Streaming," IEEE Transactions on Mobile Computing, vol. 24, no. 10, pp. 10441-10455, 2025.
[28] H. Nguyen et al., "An Accurate Viewport Estimation Method for 360 Video Streaming Using Deep Learning," EAI Endorsed Trans. on Industrial Networks and Intelligent Systems, vol. 9, no. 4, p. e2, 2022.
[29] N. V. Hung et al., "Building Predictive Smell Models for Virtual Reality Environments," Computer Science and Automation, vol. 24, no. 2, pp. 556-582, 2025.
[30] N. V. Hung et al., "HEVERL: Head-eye Movement Oriented Viewport Estimation Based on Reinforcement Learning," Journal on VR/Multimedia Systems, 2025, "Viewport Prediction for 360-

degree Video Using Reinforcement Learning," Russian Academy of Sciences, 2025.
[31] Y. Wang et al., "Synergistic Temporal-spatial User-aware Viewport Prediction for Optimal Adaptive 360-degree Video Streaming," IEEE Transactions on Broadcasting, vol. 70, no. 2, pp. 453-467, 2024.
[32] J. Li, Y. Wang and Y. Liu, "Meta360: Exploring User-specific and Robust Viewport Prediction in 360-degree Videos through Bi-directional LSTM and Meta-adaptation," Proc. of the 2023 IEEE Int. Symposium on Mixed and Augmented Reality (ISMAR), pp. 652-661, Sydney, Australia, 2023.
[33] A. Yaqoob and G.-M. Muntean, "A Combined Field-of-view Prediction-assisted Viewport Adaptive Delivery Scheme for 360° Videos," IEEE Trans. on Broadcasting, vol. 67, no. 3, pp. 746-760, 2021.
[34] Z. Li, Y. Wang, Y. Liu, J. Li and P. Zhu, "JUST360: Optimizing 360-degree Video Streaming Systems with Joint Utility," IEEE Transactions on Broadcasting, vol. 70, no. 2, pp. 468-481, 2024.
[35] C. Wu, Y. Chen, Y. Chen, F. Guo, X. Qin and H. Lu, "Physiological Signal-driven QoE Optimization for Wireless Virtual Reality Transmission," arXiv: 2508.09151, DOI: 10.48550/arXiv.2508.09151, 2025.
[36] V. H. Nguyen et al., "Retina-based Quality Assessment of Tile-coded 360-degree Videos," EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, vol. 9, no. 32, p. e2, 2022.
[37] M. Elwardy et al., "On the Consistency of 360 Video Quality Assessment in Repeated Subjective Tests: A Pilot Study," EAI Endorsed Trans. on Industrial Networks and Intell. Systems, vol. 11, no. 1, Jan. 2024.
[38] R. B. Qasem Qananwah et al., "The Utilization of EEG Signal in Video Compression," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, pp. 263-274, 2019.
[39] H. N. Viet et al., "COSMN: Clustering-based Optimization for 360-degree Live Streaming over Mobile Networks," EAI Endorsed Trans. on Industrial Networks and Intell. Sys., vol. 13, no. 1, pp. 1-16, 2025.
[40] X. Corbillon et al., "360-degree Video Head Movement Dataset," Proc. of the 8th ACM on Multimedia Systems Conf. (MMSys'17), pp. 199-204, DOI: 10.1145/3083187.3083215, 2017.
[41] D. Nguyen, N. V. Hung, T. T. Huong and T. C. Thang, "A Cross-layer Framework for Multi-user 360-degree Video Streaming over Cellular Networks," Proc. of the 2022 IEEE Int. Conf. on Consumer Electronics (ICCE), DOI: 10.1109/ICCE53296.2022.9730536, Las Vegas, USA, 2022.

ملخص البحث:

يُعدّ بثّ الفيديو بزاوية 360 درجة متعدّد الطّبقات إلى عدّة مستخدمين تحدّياً بالنّظر إلى محدودية موارد الرّاديو، واختلاف ظُروف القنوات، وتأثُّر جودة تجربة المستخدم بشكلٍ كبير بحجم شاشة العرض. ولمعالجة هذه المشكلة، نقترح في هذه الدّراسة نظاماً لتخصيص الموارد يُراعي حجم شاشة العرض لبثّ الفيديو إلى عدّة مستخدمين بزاوية 360 درجة. ويعتمد النّظام المقترح تصميماً ثنائي المراحل؛ المرحلى الأولى هي الرئيسية، وفيها يتمّ تقسيم المستخدمين إلى عددٍ من المجموعات ويخصّص النّظام لكلّ مجموعة ميزانية موارد. أمّا المرحلة الثانية، فهي مرحلة الضّبط الدّقيق، وفيها يتمّ تعظيم مقياس يُراعي جودة الصّورة بالإضافة إلى عبء الإرسال، محسِّناً بذلك من سهولة الحساب، مع الحفاظ على خدمةٍ فراغيةٍ لِمَنْفَذِ العرض في سيناريوهات تعدُّد المستخدمين بشكلٍ كثيف.

وتُظهر نتائج المحاكاة أنّ النّظام المقترح يحقّق جودة تجربةٍ تنافسية عالية عند مقارنته بأنظمةٍ أخرى مماثلة واردة في أدبيات الموضوع، مع أوقات حلٍّ أقصر بشكلٍ ملحوظ.