EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems
EAI.EU Research Article

COSMN: Clustering-Based Optimization for 360-Degree Video Streaming over Mobile Networks

Nguyen Viet Hung[1], Bui Huy Hoang[2], Tran Thanh Cong[2], Truong Thu Huong[2,*]

[1]International Training and Cooperation Institute, East Asia University of Technology, Bacninh, Vietnam
[2]School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

Abstract

The rapid growth of 360-degree video streaming has transformed how users experience immersive content, especially on mobile devices. However, delivering high-quality 360-degree video streams to mobile devices is challenging due to their constrained computational resources, limited bandwidth, and the need for real-time processing. The paper introduces COSMN (Clustering-Based Optimization for 360-Degree Video Streaming over Mobile Networks), an innovative framework to tackle these challenges. COSMN leverages a clustering-based optimization approach to dynamically adapt video streaming to the viewer's region of interest (ROI), minimizing resource consumption while maintaining high-quality visuals for the most relevant portions of the video. The framework operates by dividing the 360-degree video into multiple tiles and clustering these tiles based on user viewing patterns. By predicting user behavior with clustering algorithms, COSMN efficiently prioritizes bandwidth and processing power for the tiles within the viewer's ROI. The system also integrates adaptive bitrate streaming techniques to ensure seamless playback under varying network conditions. Experimental results demonstrate that COSMN significantly reduces bandwidth usage and computational load on mobile devices while providing a smooth and immersive viewing experience. Compared to traditional 360-degree online streaming methods, COSMN achieves superior performance in terms of latency, video quality, and resource efficiency. This work paves the way for scalable, 360-degree online streaming solutions on mobile platforms, making immersive video experiences more accessible and practical for everyday users.

Received on 07 June 2025; accepted on 10 November 2025; published on 11 November 2025

Keywords: Mobile Network, Field Of View, 360-Degree Video, Clustering-Based Optimization, COSMN, Quality of Experience

doi:10.4108/eetinis.131.9499

1. Introduction

Virtual reality (VR) technology has advanced to create highly realistic three-dimensional (3D) environments, allowing users to vividly explore a fully immersive 360 degree space. These environments are entirely computer-generated or captured from real-world scenes using 360-degree cameras, both offering a strong sense of presence and immersion. In this virtual setting, users can interact with objects and navigate spaces as if they were real. Experiencing VR typically requires specialized devices, such as virtual reality headsets or other display tools, which effectively recreate environments with remarkable realism. By providing an immersive and interactive experience, VR has found widespread application in various fields. From [1–3] education and healthcare to manufacturing and entertainment, VR has uncovered great potential, enhancing productivity, facilitating learning, and providing users with unparalleled entertainment experiences.

360-degree videos with high resolution (e.g., ≥ 4K) [4] demand substantial bandwidth while mobile devices often struggle to handle such content, leading to decreasing user experience. To overcome this challenge, there are several solutions rely on tile-based viewport adaptive streaming, unicast and multicast to optimize bandwidth usage [5–7]. Generally, 360-degree video

*Corresponding author: Truong Thu Huong. Email: huong.truongthu@hust.edu.vn

is divided into tiles, each of which is encoded with different quality levels. Within the Field of View (FoV), tiles inside viewport are distributed at high quality while tiles outside the viewport are distributed at lower quality.

Besides, Scalable High-Efficiency Video Coding (SHVC) is an extension of the H.265/HEVC compression standard, designed to optimize video compression and video content delivery in varying network conditions [8]. SHVC enables to encode video into multiple layers, consisting of base layer and enhancement layers, as illustrated in Figure 1. The base layer provides the minimum quality required to ensure content reconstruction, while the enhancement layers add information to improve resolution or overall quality. This feature allows SHVC to flexibly adjust video quality based on network conditions and available resources.

Regarding to communication bandwidth issue mentioned above, on the other hand, the advancement of mobile networks, particularly 5G, has unlocked significant potential for delivering immersive video content, including 360-degree videos. 5G networks provide extremely high bandwidth with peak data rates of up to 10 Gbps and latency as low as 1 millisecond, making them well-suited for streaming high-resolution 360-degree videos [9]. With its ability to support high-quality playback and enable advanced applications, such as live 360-degree video streaming, 5G represents a transformative solution for the future of next-generation multimedia technologies.

Therefore, the integration of high-bandwidth 5G networks and advanced video compression techniques, such as Scalable High-Efficiency Video Coding (SHVC), potentially provides a solid foundation for streaming 360-degree video content. These technologies combine with a Field of View (FoV)-based approach [10], facilitate the delivery of high-quality tiles within the user's Region of Interest (ROI) while minimizing bandwidth consumption for areas outside the viewport.

Despite this, managing network resources fairly and efficiently to avoid congestion remains a significant challenge when multiple users access 360-degree content [11] on a single 5G cell. Additionally, the simultaneous transmission of content at different quality levels is still limited in terms of optimization, as devices vary in processing and display capabilities, and users have diverse preferences regarding viewing regions.

To address such challenges, we propose a new approach to optimize 360-degree video streaming for multiple users over mobile network. Our method (called COSMN - Clustering-Based Optimization for 360-Degree Video Streaming over Mobile Networks), an optimization framework based on clustering to reduce resource consumption and ensure high-quality video delivery. COSMN leverages Scalable High-Efficiency Video Coding (SHVC) to flexibly encode various tiles. The contribution of this paper can be summarized as follows:

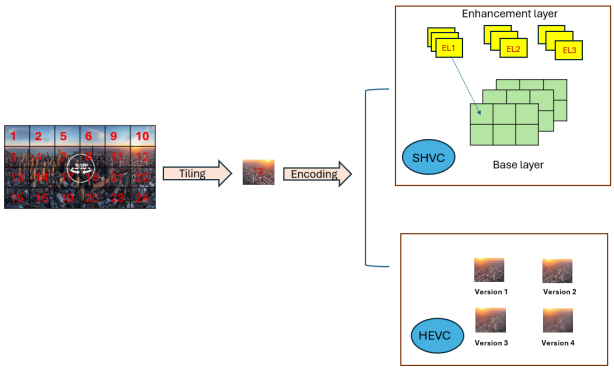


Figure 1. Video Standardization using SHVC Encoding

- By dividing 360-degree video into multiple tiles and grouping them based on clustering algorithm, COSMN prioritizes tiles in the user's region of interest (ROI) for high-quality delivery, while tiles outside the ROI can be delivered at lower quality.
- Allocating optimal resource based on constraints to improve significant Quality of Experience (QoE) compared to previous methods.

The rest of the paper is structured as follows: Section 2 discusses about the state of the art. Our proposed COSMN algorithm is presented in part 3. Next, part 5 evaluates. At the end of the paper is part 6, the conclusion.

2. Related work

Recent research on 360-degree video streaming has focused on addressing the challenges of bandwidth efficiency, scalability, and quality of experience (QoE) in mobile networks. A summary of related research on 360-degree video streaming is given in TABLE 1.

The study [12] proposes an optimization framework that dynamically adapts between unicast and multicast modes to improve QoE while minimizing bandwidth usage. Similarly, [13] introduces a cross-layer design that considers both application layer and network layer parameters to optimize resource allocation and ensure fairness in multi-user environments. Meanwhile, [6] focuses on the use of multicast solutions to improve scalability and reduce network load, allowing efficient delivery of content in 360 degrees of bandwidth intensive to multiple users simultaneously. These studies provide valuable insights and methodologies for strengthening 360-degree video streaming. However, we still find the need for further efforts to develop a unified framework that balances scalability in video

Table 1. Brief overview of related studies on 360-degree video streaming

Research	Year	Connection	Clustering	Region of Interest (ROI)	Resource Allocation	Tile Layer Scalability	QoE Measurement
LVSUM [12]	2023	Multi-user	No	No	Joint optimization-based	Yes	No
Multicast Sca [6]	2022	Multi-user	No	No	Joint optimization-based	Yes	No
Multicast All [13]	2022	Multi-user	No	No	Joint optimization-based	Yes	No
JUMPS [5]	2020	Multi-user	No	No	Joint optimization-based	No	No
Liveroi [14]	2021	Single-user	No	Yes	DL-learning based	No	No
DFT [15]	2023	Single-user	No	No	Heuristic-based	No	Yes
Tile rate allocation [16]	2020	Single-user	No	No	Knapsack-based	No	No
BBAG [17]	2024	Single-user	No	No	Heuristic-based	Yes	Yes
Clustering-Based Viewport Prediction [18]	2020	Multi-user	Yes	Yes	Viewport-aware based	No	No
Macrotile [19]	2022	Single-user	Yes	No	Heuristic-based	No	Yes
SAM [20]	2025	Multi-user	Yes	Yes	ML-learning based	No	No
Proposed COSMN	2025	Multi-user	Yes	Yes	Joint optimization-based	Yes	Yes

quality and network resource allocation, adapting to user behavior based on Regions of Interest (ROI), and finally improving QoE under diverse and dynamic network conditions.

Besides, some studies have been conducted to develop advanced streaming techniques aimed at delivering high-quality 360-degree videos with low latency and optimizing network resource usage. One of these is VAS (Viewport Adaptive Streaming) – a proposed method to reduce bandwidth requirements while still providing good quality for 360-degree videos. Due to the limited field of view, the basic principle of VAS is to only stream the viewport - the portion of the video visible to users (about 20% of the entire video) - in high quality, while the rest of the video is delivered at a lower quality. VAS approaches include viewport-dependent and tile-based methods [6]. The viewport-dependent method dynamically selects specific viewports during the streaming of 360-degree videos to reduce bandwidth usage. These areas are encoded with a higher quality compared to the regions outside the viewport. The system continuously adapts to the user's current viewport position and provides the best-suited version for their view, optimizing both video quality and bandwidth efficiency [21]. However, this method has several limitations, particularly when users abruptly change their viewing direction, which can lead to bandwidth wastage. Among state-of-the-art methods, the tile-based approach is the most frequently used technique in VAS [17, 22–24]. This method divides 360-degree videos into non-overlapping tiles and encodes each tile into multiple versions with different quality levels. The system selects the highest quality for the tiles within the user's viewport and lower quality for the tiles outside the viewport based on the user's viewport position and network bandwidth. This optimizes both the user's experience and bandwidth usage. Enhancing the efficiency of tile-based video streaming requires perfect head motion and accurate viewport estimation. With the advancement of Machine Learning (ML) and Deep Learning (DL), several studies have employed these techniques to predict viewports [25–27]. In [25], the authors introduced an online-updated viewport prediction model which mainly utilizes Convolutional Neural Networks (CNN) to extract the spatial characteristics of video frames and Long Short-Term Memory (LSTM) to learn the temporal characteristics of the user's viewport trajectories. A framework was developed that integrates Reinforcement Learning (RL) algorithms with viewport information to optimize 360-degree video streaming in viewport prediction, prefetch scheduling, and rate adaptation, as researched in [26, 28]. In [27, 29], a novel approach combining both head and eye movements of viewers to predict the viewport instead of only relying on head movements as previous methods is proposed by using LSTM to analyze the sequence of input images. However, these approaches have not addressed the impact and solutions of inaccurately predicting the viewport. In work [14] , the paper presents a Region of Interest (ROI)-based viewport prediction method for mobile 360-degree VR streaming. This method proposes the fusion of video content perception and user preference feedback (i.e., in the form of user head movement trajectory), using a 3D convolutional neural network to recognize actions and word embeddings to match video content with user preferences. Although,

this method is not a lightweight solution and the fusion of video content and user feedback may not be the best in specific scenarios. Another study [18] applies a viewer clustering method based on viewing history and viewport patterns, assigning the current user to the appropriate cluster by matching their viewport patterns with existing clusters, and predicting the viewport based on the characteristics of that cluster. Meanwhile, research [19] proposes a macrotile based 360° video streaming algorithm in which popularly viewed areas are encoded as macrotiles. The authors leverage the historical viewing data when users watch the same video, then identify the viewing centers of these users and cluster them together so that they can identify the macrotiles. In [30], the paper introduces a novel feature based viewport clustering algorithm that takes spatial, temporal, motional and other behavioral features of viewport patterns into account to characterize the user behavior in a large dataset. However, this method does not consider effective resource allocation, which affects the efficiency of processing and delivering 360° video. Similarly, the authors in [31] present a dynamic architecture that clusters users for optimized streaming based on their predicted FoV. This clustering leverages a DBSCAN-inspired algorithm that incorporates user head movement data and groups users with similar FoV preferences to enable transmission of only the relevant portions of the 360° video to each cluster. Moreover, study [20] propose a new bandwidth-aware framework that maximizes the situational awareness of a given region, using mobile digital boxes and 360° cameras, mounted on connected vehicles, taking into account the constrained uplink capacity. The proposed framework leverages the multiview spectral clustering approach and the K-Means++ algorithms to ensure efficient clustering of vehicles based on their GPS coordinates.

In mobile network environments, where challenges such as large video data volumes, unstable network conditions, bandwidth resource limitations and dynamic changes in factors like user location and device resolution are encountered, research [5, 15, 16, 32–36] have been suggested to address these issues. The authors in [35] introduced a context-aware adaptive video prefetching mechanism designed to ensure QoE-guaranteed 4K Video-on-Demand (VoD) delivery over the global Internet. This method is based on a system architecture called MVP (Mobile edge Virtualization with adaptive Prefetching), which allows content providers to integrate their content intelligence as a Virtual Network Function (VNF) within the edge infrastructure of Mobile Network Operators (MNOs). Similarly, research [32] considered a network architecture designed for mobile VR video streaming, which included a server holding the VR video content, an MVP handling the VR video packets, and a head-mounted display along with a buffer serving together as the user equipment (UE). In the context of 5G wireless network, a novel multi-path multi-tier 360° video streaming solutions was developed in study [36] . By leveraging high-throughput low-latency of 5G network, the authors proposed tile-based FoV correction and chunk retransmission schemes to tackle FoV prediction errors and late chunk deliveries. In [15, 16, 34, 37], the authors proposed a viewport adaptation based on tile-based and multicast to stream 360-degree videos to multiple users. These methods optimize the management of eMBMS (evolved Multimedia Broadcast Multicast Service) resources by dividing users into multiple multicast groups with different conditions and determining the quality of the tiles in each group. However, these solutions do not consider individual viewing preferences and ignore the independence of each tile – a factor exploited in [5, 6] to improve resource allocation and user experience. In [6], a combination Scalable Video Coding (SVC) and multicast approach incorporating Linear Regression algorithm for weight estimation of tiles was proposed to deliver popular tiles to users in a bandwidth-efficient manner. In [5], the authors recommended that each tile in a group be delivered in unicast or multicast mode to balance the diversity in user behavior. Recent research [12] introduced a framework for 360-degree video streaming that optimizes bandwidth usage through transmission between multicast and unicast. With this method, the authors integrated HEVC (High-Efficiency Video Coding) to encode the video into tiles with multiple versions and spectral efficiency was adjusted to fit individual requirements. However, this technique does not yet support scenarios where multiple users simultaneously view various video contents, including online 360° videos. Besides using HEVC encoding, study [38] proposes an ROI-based SHVC and HEVC tile method. This method uses SHVC to encode the entire 360-degree video, with the base layer (BL) at a low resolution and the enhancement layer (EL) at a high resolution for the ROI tiles. The HEVC is used so that low and high resolution sequences are separately encoded as the BL and EL of SHVC. In our research, we focus on a novel clustering-based approach to optimize 360-degree video streaming by employing a dynamic clustering strategy for the same quality level based on their relevance to the user's Region of Interest (ROI) and utilizing SHVC to encode tiles with varying quality levels.

Our main goal - COSMN - is enhancing the Quality of Experience (QoE), optimizing bandwidth usage, and ensuring scalability for multi-user scenarios in mobile network environments.

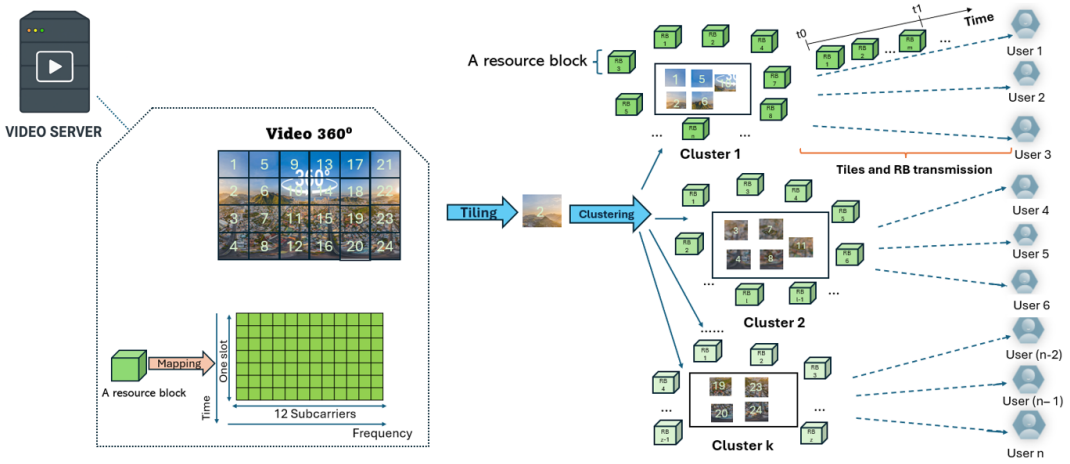


Figure 2. Resource Allocation of the Proposed System

3. SYSTEM MODEL AND PROBLEM FORMULATION

As illustrated in Figure 2, we consider a 360-degree video streaming system over mobile networks at the video server, where each 360-degree video is divided into multiple tiles and encoded into multiple quality layers (with different bitrates) using Scalable High-Efficiency Video Coding (SHVC). Additionally, each layer (version) is separated into segments of fixed duration to deliver only parts of the video within a short period. Our proposed Clustering-Based Optimization for 360-Degree Video Streaming over Mobile Networks (COSMN) dynamically allocates resources by clustering users based on their tile preferences and network conditions and assigning resource blocks (RBs) for efficient delivery in both unicast and multicast modes.

In detail, the tiles within each cluster are allocated resources based on their priority: higher-quality tiles receive more resources (i.e. RBs - Resource Blocks) than lower-quality tiles.

It is to ensure that users have a smooth experience and enjoy high-quality 360-degree video streaming in both unicast and multicast modes. When each user receives data via unicast, it increases the load on both the server and the network, particularly under fluctuating network conditions. In addition, by using Multicast for grouped users, network bandwidth and server processing capability can be utilized more effectively. Therefore, the clustering strategy potentially helps balance this by prioritizing multicast and reducing unnecessary unicast transmissions. Moreover, when multiple users receive the same data simultaneously via multicast, it significantly reduces latency and operational costs compared to multiple unicast transmissions.

The number of *RB*s required to transmit user u's version v for each tile is formulated as below:

$$RBs_{uvt} = \frac{\tau \times R_{vt}}{\sigma_u} \tag{1}$$

Where:

- RBs_{uvt}: The number of network resource blocks required to send version v of tile t to user u. A Resource Block (RBs) is the fundamental unit of bandwidth allocation in OFDM-based mobile networks. In accordance with 3GPP LTE and 5G NR standards, each RBs consists of 12 consecutive subcarriers in the frequency domain (3GPP TS 36.211; 38.211) and spans multiple time slots [39]. In this part, we adopt this standard definition to ensure consistency with current mobile network specifications. While our experiments are based on this configuration, the proposed model can be readily extended to scenarios with wider bandwidths and larger numbers of subcarriers, where the performance is expected to scale proportionally.

- σ: spectral efficiency of a communication channel. Spectrum efficiency refers to the amount of data that can be transmitted over a specific bandwidth or spectrum while minimizing transmission errors, measured in $bits/s/Hz$. Also called spectral efficiency or bandwidth efficiency, it represents the maximum number of bits of data that a cellular network can send to a set number of users per second, ensuring an acceptable level of service quality. To be more specific, the higher the spectral efficiency, the higher the throughput for the same bandwidth.

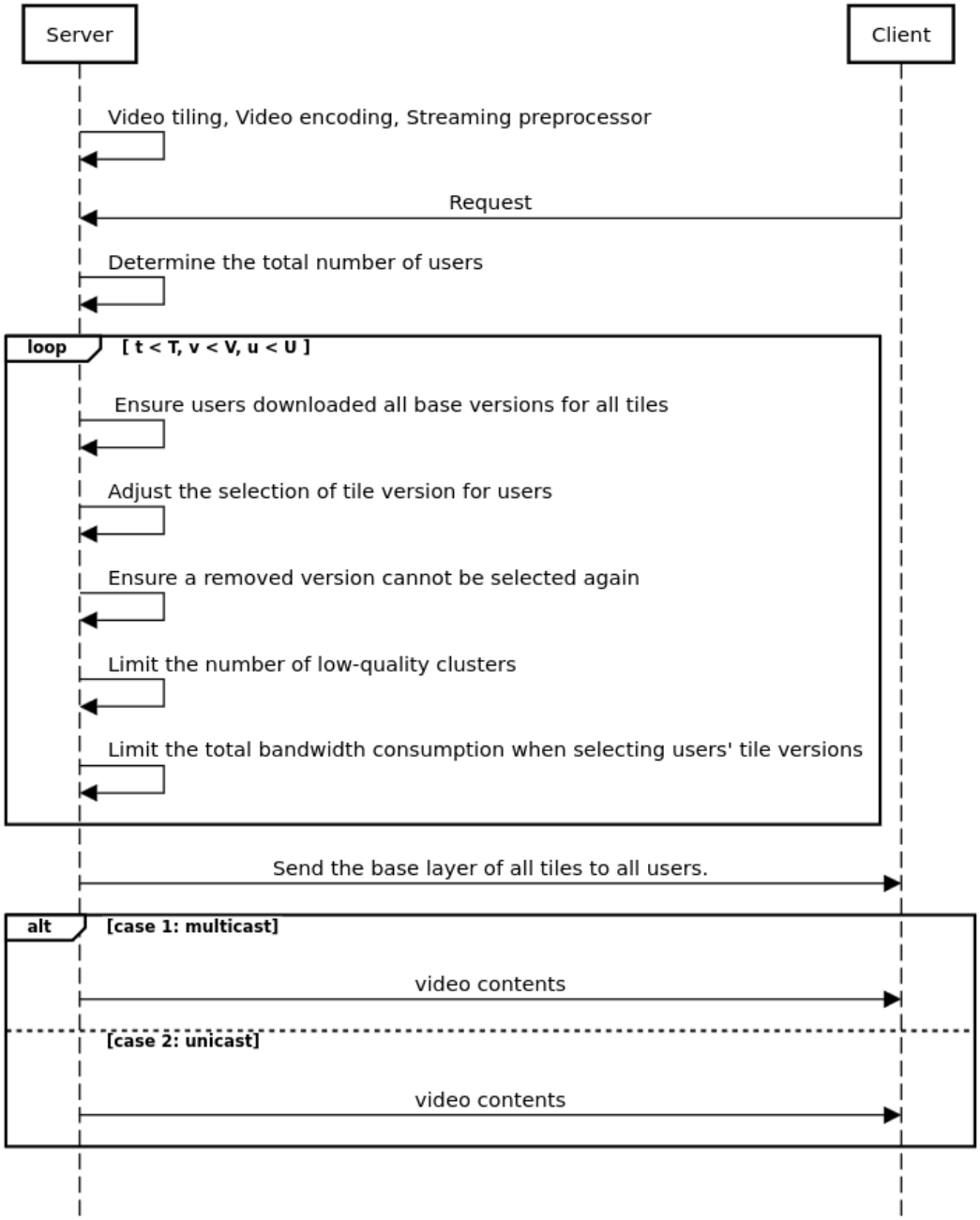


Figure 3. Overview flowchart of the COSMN system architecture

- σ_u: spectral efficiency allocated for user u, $\forall u \in [1, U], \sigma_u$ indicates spectral efficiency of the channel between the base station and the terminal device of the user.
- τ: Duration (in seconds) of each video segment to be played back.
- R_{vt}: Bitrate of video version v of tile t.

In such a system, users are grouped into C clusters according to quality preferences and channel conditions, where Cluster 1 receives the highest quality (and most resources), decreasing to Cluster C. COSMN allocates RBs dynamically over time to each cluster, depending on the tile priority (based on Region of Interest) and user link conditions.

During a time interval $[t_0, t_1]$, a user in a high-priority cluster may receive a subset of its allocated RBs and continue receiving the remaining allocation in subsequent intervals until the quota is met. This enables adaptive delivery based on instantaneous network conditions.

This dynamic resource allocation by clustering helps optimize bandwidth usage, ensuring that the tiles within ROI in 360-degree video (i.e., the tiles in regions that users are interested in, which may include several areas within a viewport with different viewing frequencies) maintain high quality, while still keeping overall bandwidth consumption efficient. Moreover, the region of interest (ROI) is defined by analyzing user behavior data, including head movement trajectories, viewport orientation changes, and the time spent on specific regions. These data are then mapped to the video's tile layout to identify which tiles fall within the user's viewing area.

Based on the above model, the resource allocation challenge can be formulated as an optimization problem aiming to maximize overall Quality of Experience (QoE), subject to bandwidth and clustering constraints.

$$\text{Maximize: } \sum_{u=1}^{U}\sum_{t=1}^{T}\sum_{c=1}^{C} w_{ut} \times Q_c \times y_{utc} \tag{2}$$

to maximize the total value of the weighted sum over all users (u), tiles (t), and clusters (c).
Where:

- w_{ut}: Weight between user u and tile t.
- Q_c: This is an array that contains the PSNR values (in dB) of each video version.
- y_{utc}: A decision variable (binary or continuous) that indicates whether user u is assigned to cluster c for tile t.
- u: Represents the user ($u \in \{1, 2, \ldots, U\}$).
- t: Represents the tile ($t \in \{1, 2, \ldots, T\}$).
- c: Represents the cluster ($c \in \{1, 2, \ldots, C\}$).

The objective function is subjected to the following constraints:

$$y_{u,t,0} = 1, \forall u \in U, \forall t \in T \tag{3}$$

$$y_{u,t,c} - y_{u+1,t,c} \leq 0 \tag{4}$$

$$z_{u+1,t,c} - y_{u,t,c} = 0 \tag{5}$$

$$\sum_{c=0}^{1} y_{u,t,c} \leq 5, \forall u \in U, \forall t \in T \tag{6}$$

$$\sum_{u=1}^{U}\sum_{t=1}^{T}\sum_{c=1}^{C}(y_{u,t,c} - z_{u,t,c}) \times (\lambda_c \times \frac{BW}{\sigma_u}) \leq R \tag{7}$$

Eq. 4 and Eq. 5 are applied $\forall u \in [1, U-1], \forall t \in [1, T]$.

Two binary variables $y_{u,t,c}$ and $z_{u,t,c}$ are defined as follows:

- $y_{u,t,c} = 1$: user u chooses cluster c for tile t;
- $y_{u,t,c} = 0$: otherwise;
- $z_{u,t,c} = 1$: cluster c is removed for user u in tile t;
- $z_{u,t,c} = 0$: cluster c is still available for user.

It is noted that these constraints are updated iteratively at each segment due to the change of users' viewing directions while watching a 360-degree video and time-varying networks. Specifically, users' throughput and viewport data are collected through feedback channel and used to recompute the feasible region of the optimization problem. Then, the system resolves the objective function with the new parameters, ensuring that COSMN adapts to time-varying network conditions while maintaining stable playback quality.

4. Proposed Method: COSMN Framework

The COSMAN system (in Figure 3) is designed to optimize 360-degree video streaming by balancing bandwidth consumption and user experience. On the server side, the video is first divided into tiles, encoded into multiple quality levels, and preprocessed for adaptive transmission. Upon receiving user requests, the system determines the total number of clients and iteratively allocates resources. During this process, it ensures that all users receive the base version of every tile to guarantee minimum video accessibility. The system then adjusts the selection of higher-quality tiles according to users' bandwidth conditions and predicted viewing areas, while preventing re-selection of removed versions. To further enhance Quality of Experience (QoE), the system limits both the number of low-quality clusters and the overall bandwidth usage per user. Finally, the server delivers the base layer of all tiles to clients and distributes enhanced versions via multicast or unicast depending on the network scenario. This architecture enables efficient bandwidth management while maintaining a consistent viewing quality across multiple users.

Overall, the whole process of the proposed COSMN method is summarized, being divided into 3 main steps:

- Step 1: Content preparation and streaming pre-processing
- Step 2: User clustering based on network conditions
- Step 3: Selection between unicast and multicast transmission

Step 1: Content preparation and streaming preprocessing

The server performs content preparation, which consists of video tiling, video encoding, and streaming

preprocessing tasks. On this side, all versions of a given 360-degree video are stored together with the bitrates for each version.

In the proposed system, the server and client are connected within the same Local Area Network (LAN) to ensure low transmission latency. The client receives the video content from the server through a transmission channel, while a feedback channel conveys information such as the current status of the client's network and viewport data to the server. The playback of a segment starts only after all its tiles have been successfully downloaded at the client.

The facet is that, while viewing 360-degree video, users can only see a portion of the video that is called viewport [40], due to the limited Field-of-View (FoV).

Therefore, in the COSMN framework, we deploy a tiling-based technique in the Video Adaptive Streaming (VAS) system to divide a 360-degree video into multiple tiles. VAS only delivers high-bitrate video segments to the viewport area, which the user is mainly looking at, while delivering the other parts of the video with lower video quality. This technique helps reducing unnecessary data consumption as well as ensure that when user suddenly turns their head, they can still see content rather than experiencing a blank. Thus, bandwidth demand for delivering a 360-degree video can be reduced.

Then, before entering the second step, users send requests to the server, allowing it to determine the total number of users.

Step 2: User clustering based on network conditions

In the second step, users are clustered according to network conditions

As shown in Figure 4, a 360-degree video is divided into T tiles. Each tile is assigned a specific weight w_{ut} where w_{ut} represents the weight between the user and the tile, which is calculated as the ratio between the visible pixels of tile t and the total number of pixels in the viewport. Tiles with higher w_{ut}, meaning tiles located within the ROI (Region of Interest) or receiving more user attention, are prioritized for early transmission to the user compared to tiles nearby the ROI.

Subsequently, U users sharing the same quality level Q_c (or sharing the same version) are grouped into C clusters. Each cluster may have a different number of users, depending on the quality level corresponding to each user. The goal is for the tiles to be transmitted to each cluster with different versions suitable for their current quality levels. This delivery process follows a timeline: when user u in cluster c, which has quality level Q_{cC}, wants to download version v of tile t (associated with a specific weight w_{ut}), the system sequentially transmits versions from version 0 up to version (v-1) during the user's download process.

For example, as illustrated in Figure 4, assuming tiles 8, 7, and 4 have descending weights ($w_{ut8} > w_{ut7} > w_{ut4}$) and are transmitted to two clusters, C and (C-1) during an interval from t_0 to t_{12}. Each tile is encoded into 5 versions, $\forall v \in [0, 4]$ with a base version (v_0) and four enhancement versions (v_1 to v_4).

Cluster c consists of 3 users having higher quality ($Q_{cC} > Q_{c(C-1)}$), while cluster (c-1) has 4 users sharing lower-quality tile versions. Tile 8, which is located in the center of the ROI, has the highest weight and is transmitted first with version v_4 from t_0 to t_5, followed by tile 7 from t_5 to t_9, and tile 4 from t_9 to t_{12}. The delivery order and version selection of tiles depend on their w_{ut} weights and the quality levels of clusters c and (c-1). Additionally, since not all users start viewing at the same time, new users can join an ongoing streaming session at any segment and they are assigned to a suitable cluster according to their current viewport and network status. Users within a cluster can be located same spatial region or distributed through coverage area, as long as they share similar viewport patterns and network characteristics.

In general, the COSMN framework must ensure that all users have downloaded the base version of every tile. This guarantees that when a user suddenly turns their head, there is still some content available to display— even if it is of lower quality. Therefore, Eq. 3 ensures that all users have preloaded the base versions of all tiles onto their headsets required by the layered architecture of SHVC.

On the other hand, COSMN prioritizes clustering the versions selected by different users for the same tile. This concept is implemented through Eq. 4 or **Constraint 1** and Eq. 5 or **Constraint 2**, which are designed to minimize discrepancies in the tile versions received by different users. In particular, these two constraints achieve this by enforcing continuity and preventing the reuse of dropped clusters.

Algorithm 1 Adjust the selection of tile version for users

```
for tile = 0 to T − 1 do
    for cluster = 0 to C − 1 do
        for user = 0 to U − 1 do
            if user + 1 < U then
                y_{u,t,c}[user][tile][cluster] − y_{u,t,c}[user + 1][tile][cluster] ≤ 0
            else
                y_{u,t,c}[user][tile][cluster] = 0
            end if
        end for
    end for
end for
```

On the one hand, constraint 1 or Eq. 4 describes the allocation of video versions, ensuring that if current

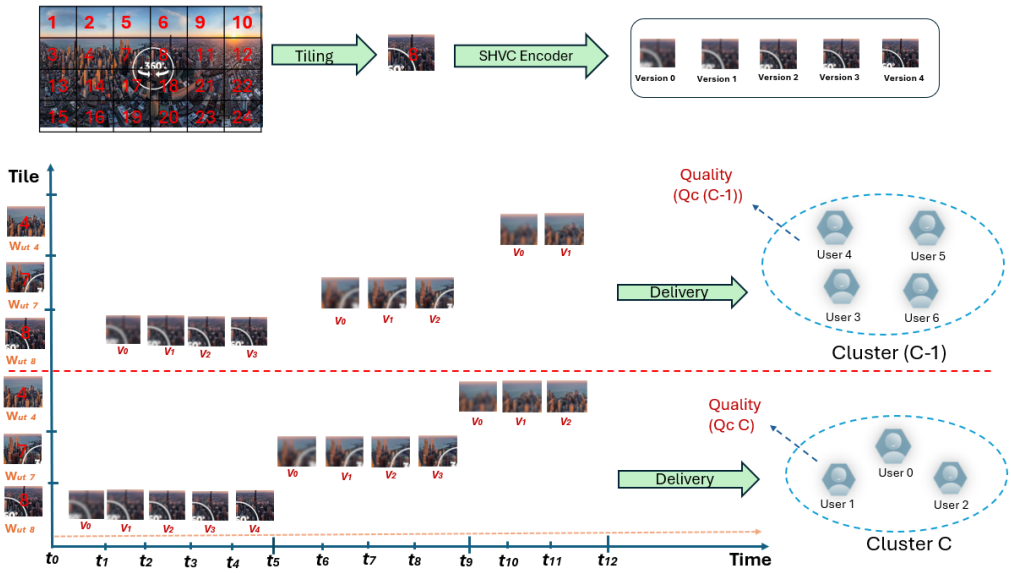


Figure 4. Tile Version Clustering and Delivery

user u_n is not assigned a specific video version v, all subsequent users (u_{n+1}, u_{n+2}, u_{n+3}, ...) can not select video version v either. Contrast, if user u_n in cluster c receives version v of tile t, all users from user u_{n+1} to user U also receive tile version. This creates a consecutive group of users selecting the same video version, which is how our approach groups them into a cluster to enable multicast mode.

For instance, if there are 5 users (denoted as u_1, u_2, u_3, u_4, u_5) respectively and the system is trying to group them into a cluster to send video version v_3. So if u_1, u_2, u_3 is chosen to send version v_3 but u_4 is not sent, u_4 and u_5 have to switch to the other video version instead of the same version v_3 . Then, u_1, u_2, u_3 are grouped and video data is delivered via multicast.

Algorithm 2 Ensure a removed version cannot be selected again

```
for tile = 0 to T − 1 do
    for cluster = 0 to C − 1 do
        z_{u,t,c}[0][tile][cluster] = 0
        for user = 0 to U − 1 do
            if user + 1 < U then
                z_{u,t,c}[user + 1][tile][cluster] −
                y_{u,t,c}[user][tile][cluster] = 0
            end if
        end for
    end for
end for
```

On the other hand, constraint 2 ensure the video tile version which is removed can not be able to access again. This algorithm supports the previous one in the case of grouping of users to deliver via multicast.

This approach ensures a more uniform viewing experience by user receiving an equally delivered tile version. For instance, it prevents scenarios in which one user receives the highest-quality version of a tile while others receive much lower-quality versions. Additionally, by promoting the selection of identical tile versions among users, the system significantly reduces the number of tile versions that must be delivered—similar to users, like what occurs in unicast transmission mode.

This multicast mode can significantly improve bandwidth efficiency, and in turn enhancing the overall Quality of Experience (QoE) by ensuring consistent video quality delivery, especially under constrained network conditions.

Nonetheless, by incorporating Eq. 6 into the optimization process, the system effectively prevents over-prioritization of low-quality versions, which could otherwise degrade the overall QoE. The equation introduces a weighting mechanism that balances the delivery of high-quality tiles with efficient bandwidth consumption. In this context, the value of 5 is chosen as a design parameter to control the steepness of the weighting curve. A smaller value would reduce the prioritization of higher-quality versions, whereas a larger value would excessively penalize low-quality versions. Our experimental results show that setting this parameter to 5 achieves a desirable trade-off, aligning with the system's goal of maintaining QoE while managing bandwidth. Although the parameter can be tuned to meet different system requirements, the default choice of 5 has proven effective in practice.

Eq. 7 is formulated to constrain total bandwidth usage during the selection of tile versions for users,

Table 2. The bitrate and quality of each video will be considered in our work

Videos			Version 0	Version 1	Version 2	Version 3	Version 4
"Less feature" videos	Roller Coaster	PSNR (dB)	39.45	42.86	44.99	47.28	49.42
		Bitrate (kbps)	54.86	131.71	250.53	515.88	933.84
	Venice	PSNR(dB)	32.73	36.07	38.69	41.76	44.76
		Bitrate (kbps)	60.07	183.02	384.87	826.38	1520.55
	Paris	PSNR (dB)	38.20	42.19	44.98	47.88	50.72
		Bitrate (kbps)	60.78	130.38	209.69	340.25	532.44
"More feature" videos	Rhino	PSNR (dB)	36.10	39.19	41.93	45.40	48.11
		Bitrate (kbps)	196.66	441.00	731.52	1234.01	1787.97
	Diving	PSNR (dB)	34.49	38.22	40.97	44.01	46.57
		Bitrate (kbps)	158.15	393.63	752.26	1481.51	2499.02

ensuring that the overall resource consumption does not exceed the available network capacity at any given moment. The equation includes several key parameters:

- BW: the bandwidth required to download each version or it is actually the bitrate of each video segment after encoding.
- λ_c: the priority weight of each tile version.
- σ_u: The spectral efficiency of the connection between each user and the base station.
- R : The available resource blocks or the available network resources.

This equation plays a crucial role in maximizing QoE by avoiding network congestion and ensuring efficient resource allocation. By leveraging this equation, the system can support a large number of users while maintaining optimal performance.

Step 3: Selection between unicast and multicast transmission

In the final step, the system performs the transmission, which begins only after Step 2 has completed. COSMN delivers video versions to users for their tiles using two delivery modes: unicast and multicast.

- **In multicast mode**: Users selecting the same version of a tile are grouped into clusters to efficiently receive data via multicast. This selection is enforced by optimization constraints, which take into account the network conditions of all users within the same cluster. The assigned version is determined by considering the minimum available bandwidth among users in the cluster to ensure successful delivery to every member. In our work, we assume that there are n given Resource Blocks (RBs) and the system determines U users to form the optimal C clusters for multicast transmission. The number of users in each cluster (denoted as M), where $M \geq 2$, can vary depending on factors such as quality level and the network conditions of individual users.
- **In unicast mode**: This mode is activated when users cannot be grouped together because each user receives a different version of a tile.

Essentially, our approach prioritizes the multicast strategy to minimize the needed bandwidth compared to the unicast mode in which the bandwidth is demanded by each individual user. By delivering the same video versions to multiple users simultaneously in multicast mode, overall bandwidth consumption can be significantly reduced. Moreover, the versions can be upgraded or downgraded across segments depending on changes in network conditions and remaining bandwidth of users in each cluster. Therefore, this approach is particularly effective in delivering 360-degree video content to multiple users with similar regions of interest (ROI) during the same time period. Consequently, our method not only optimizes the use of network resources, but also improves the overall quality of experience (QoE).

5. Performance Evaluation

5.1. Experimental Settings

Our performance evaluation is conducted through a custom-built simulator implemented in Python. We use the dataset of [41], which consists of five 360-degree videos: Rollercoaster, Diving, Venice, Paris and Rhino, which stands for "more feature" and "less feature" categories, with detailed specifications provided in Table 2. The settings for the two categories reflect a diverse range of real-world video conditions, where the content may include various bitrates and qualities—ranging from dynamic scenes with significant motion to mostly static content—providing a fairer basis for comparison. To be more specific, "more feature" videos (i.e. Rhino video and Diving video) contain dynamic elements, including numerous moving objects and complex camera movements. In contrast, the "less feature" videos show almost static video, which means that almost all objects in the video

Table 3. The bitrate and quality used in the experiments

Videos		Version 0	Version 1	Version 2	Version 3	Version 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

are static or just moving slowly. Overall, "more feature" videos have higher bitrates compared with "less feature" videos. However, in terms of quality calculated in dB, "less feature" videos have a higher quality viewport according to our formula. The current viewport for each user is randomly generated as a viewport trace. Moreover, all videos in the evaluated dataset have frame rates ranging from 30 to 40 fps.

In addition, Table 2 shows the average bitrate (in kbps) and the corresponding tile quality, measured in Peak Signal-to-Noise Ratio (PSNR) for each version of each video. PSNR is a commonly used objective fidelity metric that measures the pixel-wise difference between the original and compressed video, expressed in decibels (dB). In our system, the mean of each bitrate and viewport qualiAty for each version is computed, illustrated in Table 3.

The videos are projected using the Equirectangular Projection (ERP) format, chosen for its simplicity compared to other projection methods [42], and are converted to a resolution of 2890×1920 pixels. Each frame is partitioned into 24 tiles with a resolution of 480×480 pixels per tile. To support scalable video streaming, each tile is encoded using the Scalable High Efficiency Video Coding (SHVC) extension of the HEVC standard [43], consisting of one base layer and four enhancement layers.

Fixed quantization parameters of 38, 32, 28, 24, and 20 are applied for the Base layer, Enhancement Layer (or video version) 1, 2, 3, and 4, respectively, as proposed in the work [17].

The spectral efficiency (σ_u) of the users is computed under the setting of U of 45 and 60 corresponding to 45 and 60 users, the same as used in work [12]. The priority weight λ_c for each tile version are 0.7, 0.8, 0.7, 1.0 and 1.0 for Base layer, Enhancement Layer 1 through 4, respectively.

The proposed method is tested using Gurobi 12.0.0 Solver[1], which is performed on a 64-bit Windows 10 laptop with 20GB Memory and 1.19GHz Intel core i5 CPU.

For performance analysis, the three following reference methods are used for comparision, as outlined below:

- LVSUM [12]: This method transmits viewport tiles in multiple enhancement layers for maximum quality, while off-viewport tiles use only the base layer to save bandwidth. Unlike traditional methods, LVSUM combines unicast and multicast modes, dynamically optimizing network resources to improve video quality and spectral efficiency for multiple users.
- Multicast Sca [6]: This method combines Scalable Video Coding and multicast transmission to optimize 360-degree video delivery, with base layers multicasted to all users and enhancement layers transmitted via unicast or multicast as needed.
- Multicast All [13]: This is a simplified version of our proposed method, where all users are limited to receiving the same set of tile layers.

In order to verify whether COSMN outperforms in terms of QoE score, we have examined various QoE calculation methods proposed in previous studies, including [44–46]. However, the QoE computations in these studies are applicable only to 2D environments. Additionally, some studies, such as [7, 17, 47–51], specifically focus on 360-degree videos. Among them, the QoE formulation from [47] is widely adopted and is defined as follows:

$$QoE = \sum_{i=0}^{S} (\alpha \times bitrate_i - \beta \times rebuffer_i - \gamma \times smooth_i) \tag{8}$$

In this formulation, higher QoE can be achieved by delivering higher bitrate, reducing rebuffer, and minimizing smooth variation between consecutive segments, which helps reduce jitter and ensure stable playback.

Where:

- α, β and γ are assigned to be 1, 1.85 and 1, respectively, as proposed in work [47].
- S represents the total number of segments. It is noted that segment 0 ($i = 0$) refers to the initial segment index, which is considered before the playback of the first segment ($i = 1$).

[1] https://www.gurobi.com/

- $bitrate_i$ is the viewport bitrate value of segment i. If a chunk has d ownload time greater than the buffer size at the beginning of downloading the chunk, a rebuffering event occurs.
- $rebuffer_i$ value represents the time difference between the current buffer size (B_i) and the duration of the video segment (τ_{seg}), as in Equation (9).

$$rebuffer_i = \begin{cases} \left|B_i - \tau_{\text{seg}}\right|, & (B_i > \tau_{\text{seg}}) \\ 0, & (B_i = \tau_{\text{seg}}) \end{cases} \quad (9)$$

.

- $smooth_i$ is defined as the inter-segment viewport bitrate variation, which is calculated as the difference in viewport bitrate of consecutive segments, shown in Eq.(10).

$$smooth_i = |r_{i+1} - r_i| \quad (10)$$

Besides, Viewport Quality, which is calculated in dB according to the average viewport PSNR, is computed by using formula from our previous work [12], as follows:

$$AVSNR = \frac{1}{U} \sum_{u=1}^{U} AVSNR_u \quad (11)$$

$$AVSNR_u = \sum_{t=1}^{T} \Lambda_{ut} \times \Theta_{tc_u} \quad (12)$$

In Equation 12, Λ_{ut} represents the quality of the numbers on all display tiles, and Θ_{tc} represents the quality of cluster c of tile t. Additionally, the largest tile layer t is transmitted to user u, denoted c_u.

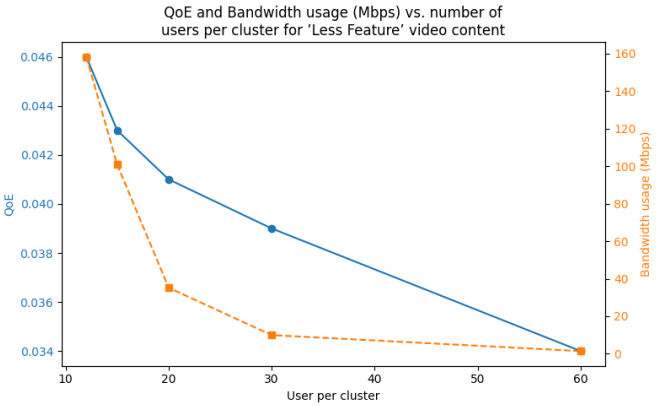


Figure 5. Quality of Experience (QoE) and Bandwidth Usage vs. number of users for Different Clusters for 'Less Feature' Video content at 100 kRBs

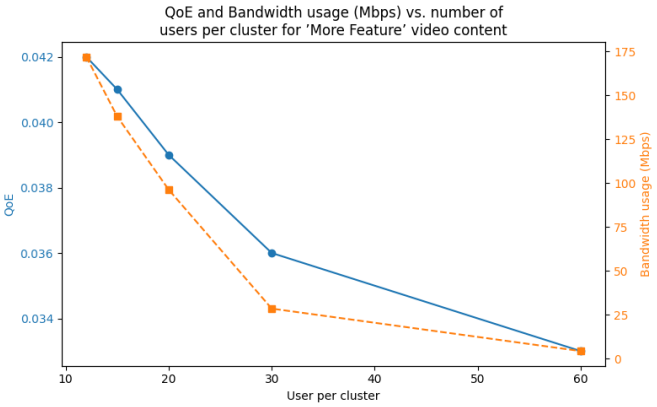


Figure 6. Quality of Experience (QoE) and Bandwidth Usage per User for Different Clusters for 'More Feature' Video content at 100 kRBs

5.2. Performance Evaluation

Impact of Clustering on QoE and Bandwidth. It should be noted that in this experiment, the system will group users into clusters with an equal number of users in each cluster. For instance, if there are 20 users per cluster, there will be 3 clusters; or if there are 12 users per cluster, the system will group users and separate them equally into all video layer clusters. One more thing to note is that the layers of the clusters in each of the following cases are randomly adapted based on the users' conditions. As a result, the video versions used in each case will vary. This experiment aims to demonstrate how grouping users into fixed numbers of clusters affects the evaluation. In our system, users will be grouped based on their network conditions and viewport, so the number of users in each cluster may vary.

In our proposed strategy, the system clusters users requesting the same video version and delivers video content to each cluster in multicast mode. Therefore, it is necessary for us to investigate the benefit of our clustering mechanism in terms of QoE and Bandwidth usage over different scenarios. Figure 5 and 6 show how grouping users into groups will affect QoE and Bandwidth usage. In our setup, each raw video is encoded into five different versions, allowing for a maximum of five clusters.

In the experiment, we set up 60 users who are grouped into clusters based on available conditions. The number of users per cluster varies depending on the network conditions and is expected to yield higher performance results, as shown in Table 4 or 5. As Figures 5 and 6 show, the more groups the system can form, the better the QoE results. However, this improvement comes with a trade-off in bandwidth usage. Figures 5 and 6 demonstrate that when only a single cluster is formed, all users receive only the base

Table 4. QoE evaluation for the "less feature" video content, under constraints of limited Resource Blocks (kRBs)

# Users	kRBs	COSMN	LVSUM	Multicast All	Multicast Sca
45	20	1,924	1,917	-	-
	30	1,936	1,927	-	-
	45	1,947	1,943	-	-
	55	1,954	1,949	-	-
	65	1,960	1,957	-	-
	75	1,965	1,962	1,873	1,880
	80	1,967	1,964	1,880	1,891
60	20	2,584	2,577	-	-
	30	2,596	2,586	-	-
	45	2,607	2,602	-	-
	55	2,613	2,609	-	-
	65	2,620	2,616	-	-
	75	2,624	2,621	2,532	2,540
	80	2,626	2,623	2,540	2,550

Table 5. QoE evaluation for the "more feature" video content, under constraints of limited Resource Blocks (kRBs)

# Users	kRBs	COSMN	LVSUM	Multicast All	Multicast Sca
45	50	1,870	1,858	-	-
	80	1,889	1,879	-	-
	120	1,902	1,896	-	-
	180	1,919	1,915	-	-
	250	1,929	1,926	1,828	1,838
	280	1,931	1,929	1,835	1,849
	300	1,931	1,930	1,840	1,855
60	50	2,515	2,503	-	-
	80	2,533	2,524	-	-
	120	2,547	2,541	-	-
	180	2,564	2,560	-	-
	250	2,573	2,571	2,473	2,483
	280	2,575	2,574	2,480	2,494
	300	2,576	2,575	2,484	2,500

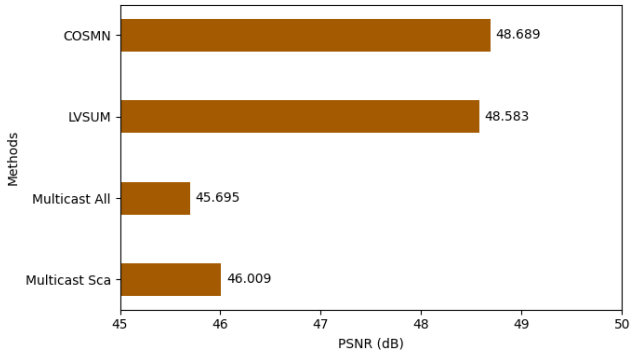
version of the video. This leads to lower bandwidth consumption, therefore resulting in a lower QoE.

Comparative QoE Analysis. Tables 4 and 5 present the QoE results measured using four different methods: COSMN, LVSUM, Multicast All, and Multicast Sca, for scenarios with 45 and 60 users under various fixed network conditions. A clear trend that can be observed: Multicast All and Multicast Sca perform poorly under low network constraints in both cases—whether the videos are of 'Less Feature' or 'More Feature' types. Moreover, under limited network conditions, our proposed method generally achieves better QoE compared to the other approaches. As mentioned in the previous section, by using our clustering proposed method, the number of users in each cluster may vary leading to the much better result as shown in Tables 4 and 5. Even when the available network resources are limited, the system can deliver superior QoE compared to the other methods.

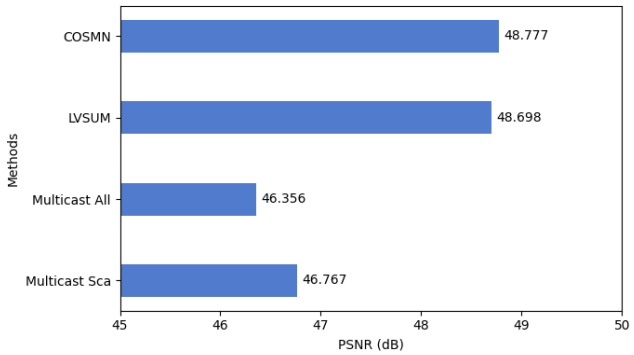
For example, in the case of 60 users watching 'Less Feature' videos under 20 and 30 *kRBs*, the differences between our work and LVSUM method is 0.007 and 0.010 unit, respectively. However, this does not imply that our method works under any level of network condition. The approach remains effective as long as sufficient resources are available to deliver at least the lower-quality version of the video.

In contrast, for 'More Feature' videos, the system requires higher available *kRBs* to maintain optimal performance, especially when grouping users to maximize efficiency.

As the number of users and clusters increases, our proposed solution demonstrates even greater advantages. In real-world scenarios involving thousands of users, the incremental improvements in QoE are expected to scale significantly. As shown in Tables 4 and 5, QoE values consistently improve with the growth in user numbers. This indicates that our method becomes increasingly effective as the user base expands, ensuring stable performance even under high network load.



(a) 45 users of 'Less Feature' videos

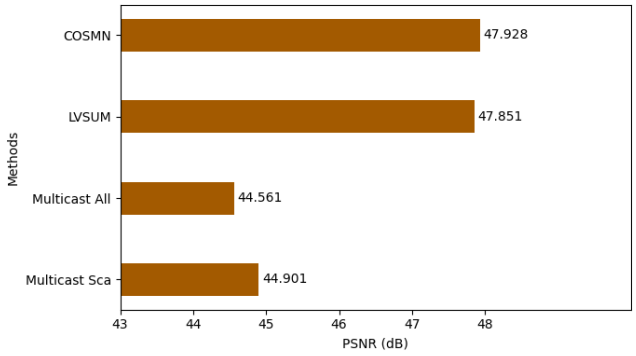


(b) 60 users of 'Less Feature' videos

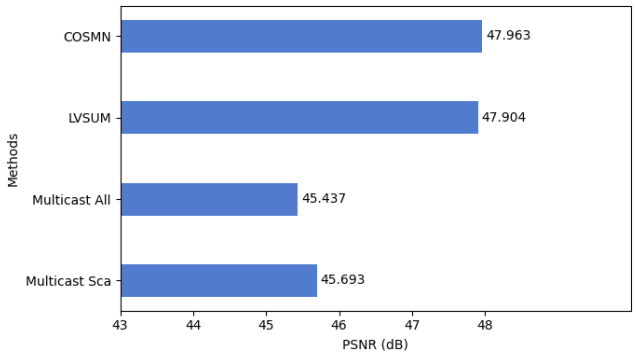
Figure 7. The performance between the proposed methods and the others (RBs = 80 kRBs) in terms of Viewport Quality (dB)

In other words, as user demand grows, our approach continues to deliver enhanced QoE, helping the system remain efficient and reliable under greater pressure.

Viewport Quality (PSNR) Evaluation. As shown in Figures 7 and 8, LVSUM and COSMN outperform the other methods. Across all evaluated scenarios, the performance ranking remains consistent, from lowest to highest: Multicast All, Multicast Sca, LVSUM, and COSMN. In addition, the difference in performance



(a) 45 users of 'More Feature' videos



(b) 60 users of 'More Feature' videos

Figure 8. The performance between the proposed methods and the others (RBs = 250 kRBs) in terms of Viewport Quality (dB)

between LVSUM and COSMN is approximately 0.1 dB in both cases. Especially, the overall results for the 'Less Feature' video content are higher compared to those for the 'More Feature' content. This can be explained by the fact that, at the input stage, the viewport quality of 'Less Feature' videos is already superior than that of the 'More Feature' videos. Therefore, the quality of the input data plays a crucial role in content delivery.

6. Conclusions

In this paper, we have presented a new approach – COSMN – that addresses the challenges associated with streaming 360-degree video over mobile networks. Through the use of Scalable High-Efficiency Video Coding (SHVC), COSMN leverages a layered encoding structure with base layer (BL) and enhancement layer (EL) that allows for adaptive video quality based on available network resources. This method facilitates flexible scalability and ensures efficient resource allocation under diverse and changing network conditions. Therefore, as the experimental results show, COSMN significantly improves Quality of Experience (QoE) by efficiently allocating resources based on users' regions of interest (ROI), through clustering users with the same quality level of tiles within each cluster. Moreover, the model ensures that bandwidth constraints are respected while maintaining high video quality for the most relevant content.

In terms of future work, a potential direction for future research is to incorporate user feedback parameters, such as user interaction frequency and viewport switching rate, as these factors can help the system more accurately identify regions of interest (ROI) and allocate resources more efficiently. Due to the challenges of collecting and processing real-time feedback data from multiple users simultaneously, these parameters have not yet been considered in this work. Furthermore, some other factors could be considered for next research, such as varying priority levels across video content or the integration of viewport estimation to optimize the clustering strategy.

Acknowledgment

This research is funded by Hanoi University of Science and Technology - HUST under Project number T2024-PC-048

References

[1] A. J. Nair, S. Manohar, A. Mittal, and R. Chaudhry, "Unleashing digital frontiers: Bridging realities of augmented reality, virtual reality, and the metaverse," in The Metaverse Dilemma: Challenges and Opportunities for Business and Society. Emerald Publishing Limited, 2024, pp. 85–112.

[2] K. Logeswaran, S. Savitha, P. Suresh, K. Prasanna Kumar, M. Gunasekar, R. Rajadevi, M. Dharani, and A. Jayasurya, "Unifying technologies in industry 4.0: Harnessing the synergy of internet of things, big data, augmented reality/virtual reality, and blockchain technologies," Topics in Artificial Intelligence Applied to Industry 4.0, pp. 127–147, 2024.

[3] M. Sayyed, B. R. Jadhav, V. Barnabas, and S. K. Gupta, "Human-machine interaction in the metaverse: A comprehensive review and proposed framework," Impact and Potential of Machine Learning in the Metaverse, pp. 1–28, 2024.

[4] J. Tu, C. Chen, Z. Yang, M. Li, Q. Xu, and X. Guan, "Pstile: Perception-sensitivity-based 360° tiled video streaming for industrial surveillance," IEEE Transactions on Industrial Informatics, vol. 19, no. 9, pp. 9777–9789, 2023.

[5] A. Majidi and A. H. Zahran, "Optimized joint unicast-multicast panoramic video streaming in cellular networks," in 2020 IEEE 28th International Conference on Network Protocols (ICNP). IEEE, 2020, pp. 1–6.

[6] 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. 38 802–38 812, 2022.

[7] V. H. Nguyen, N. N. Pham, C. T. Truong, D. T. Bui, H. T. Nguyen, and T. H. Truong, "Retina-based

quality assessment of tile-coded 360-degree videos," EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, vol. 9, no. 32, 2022.
[8] C.-H. Yeh, J.-R. Lin, M.-J. Chen, C.-H. Yeh, C.-A. Lee, and K.-H. Tai, "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.
[9] M. J. Mohammed, A. Ghazi, A. M. Awad, S. I. Hassan, H. M. Jawad, K. M. Jasim, and M. A. Nurmamatovna, "A comparison of 4g lte and 5g network cybersecurity performance," in 2024 35th Conference of Open Innovations Association (FRUCT). IEEE, 2024, pp. 452–464.
[10] F. Duanmu, E. Kurdoglu, S. A. Hosseini, Y. Liu, and Y. Wang, "Prioritized buffer control in two-tier 360 video streaming," in Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, 2017, pp. 13–18.
[11] F. Duanmu, E. Kurdoglu, Y. Liu, and Y. Wang, "View direction and bandwidth adaptive 360 degree video streaming using a two-tier system," in 2017 IEEE International Symposium on Circuits and Systems (ISCAS), 2017, pp. 1–4.
[12] N. V. Hung, P. H. Thinh, N. H. Thanh, T. T. Lam, T. T. Hien, V. T. Ninh, and T. T. Huong, "Lvsum-optimized live 360 degree video streaming in unicast and multicast over mobile networks," in 2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, 2023, pp. 29–34.
[13] 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," in 2022 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2022, pp. 1–3.
[14] X. Feng, W. Li, and S. Wei, "Liveroi: region of interest analysis for viewport prediction in live mobile virtual reality streaming," in Proceedings of the 12th ACM Multimedia Systems Conference, 2021, pp. 132–145.
[15] A. Yaqoob and G.-M. Muntean, "Advanced predictive tile selection using dynamic tiling for prioritized 360 video vr streaming," ACM Transactions on Multimedia Computing, Communications and Applications, vol. 20, no. 1, pp. 1–28, 2023.
[16] P. K. Yadav and W. T. Ooi, "Tile rate allocation for 360-degree tiled adaptive video streaming," in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 3724–3733.
[17] V. H. Nguyen, D. T. Bui, T. L. Tran, C. T. Truong, and T. H. Truong, "Scalable and resilient 360-degree-video adaptive streaming over http/2 against sudden network drops," Computer Communications, vol. 216, pp. 1–15, 2024.
[18] A. T. Nasrabadi, A. Samiei, and R. Prakash, "Viewport prediction for 360 videos: a clustering approach," in Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, 2020, pp. 34–39.
[19] X. Chen, T. Tan, and G. Cao, "Macrotile: Toward qoe-aware and energy-efficient 360-degree video streaming," IEEE Transactions on Mobile Computing, vol. 23, no. 2, pp. 1112–1126, 2022.
[20] O. El Marai, S. Messinis, N. Doulamis, T. Taleb, and J. Manner, "Roads infrastructure digital twin: Advancing situational awareness through bandwidth-aware 360° video streaming and multi-view clustering," IEEE Open Journal of Vehicular Technology, 2025.
[21] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications," in 2016 IEEE International Symposium on Multimedia (ISM). IEEE, 2016, pp. 583–586.
[22] W. Gao, C. Li, H. Lv, W. Dai, J. Zou, H. Xiong, X. Pan, and H. Wang, "Optimal tile-based encoding for 360-degree video streaming," in 2022 Picture Coding Symposium (PCS). IEEE, 2022, pp. 295–299.
[23] H. Nguyen, "Mellifluous viewport bitrate adaptation for 360 videos streaming over http/2," Journal on Information Technologies & Communications, vol. 2024, no. 2, pp. 2–2, 2024.
[24] N. V. Hung, N. A. Quan, N. Tan, T. T. Hai, D. T. Trung, L. M. Nam, B. T. Loan, and N. T. T. Nga, "Building predictive smell models for virtual reality environments," _, vol. 24, no. 2, pp. 556–582, 2025.
[25] S. Peng, J. Hu, H. Xiao, S. Yang, and C. Xu, "Viewport-driven adaptive 360○ live streaming optimization framework," Journal of Networking and Network Applications, vol. 1, no. 4, pp. 139–149, 2022.
[26] Z. Jiang, X. Zhang, Y. Xu, Z. Ma, J. Sun, and Y. Zhang, "Reinforcement learning based rate adaptation for 360-degree video streaming," IEEE Transactions on Broadcasting, vol. 67, no. 2, pp. 409–423, 2020.
[27] N. Hung, T. Lam, T. Binh, A. Marshal, and T. Huong, "Efficient deep learning-based viewport estimation for 360-degree video streaming," Advances in Science, Technology and Engineering Systems Journal, vol. 9, pp. 49–61, 05 2024.
[28] W. Feng, S. Wang, and Y. Dai, "Adaptive 360-degree streaming: Optimizing with multi-window and stochastic viewport prediction," IEEE Transactions on Mobile Computing, pp. 1–14, 2025.
[29] N. Hung, P. Dat, N. Tan, N. Quan, L. Trang, L. Nam et al., "Heverl–viewport estimation using reinforcement learning for 360-degree video streaming," Informatics and Automation, vol. 24, no. 1, pp. 302–328, 2025.
[30] A. Dharmasiri, C. Kattadige, V. Zhang, and K. Thilakarathna, "Viewport-aware dynamic 360 {\deg} video segment categorization," arXiv preprint arXiv:2105.01701, 2021.
[31] A. Saadallah, S.-M. Senouci, I. El-Korbi, and P. Brunet, "Dynamic field-of-view-based clustering for efficient 360-degree multicast streaming," in GLOBECOM 2024-2024 IEEE Global Communications Conference. IEEE, 2024, pp. 602–607.
[32] T. M. C. Chu and H.-J. Zepernick, "Performance analysis of an adaptive rate scheme for qoe-assured mobile vr video streaming," Computers, vol. 11, no. 5, p. 69, 2022.
[33] N. H. Lich, T. T. Huong, N. V. Hung, and P. N. Nam, "Efficient short-form video streaming: An integration of dynamic bitrate adaptation and predictive segment preloading," in 2024 Fifteenth International Conference

on Ubiquitous and Future Networks (ICUFN). IEEE, 2024, pp. 360–365.
[34] 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, 2017, pp. 170–178.
[35] C. Ge, N. Wang, G. Foster, and M. Wilson, "Toward qoe-assured 4k video-on-demand delivery through mobile edge virtualization with adaptive prefetching," IEEE Transactions on Multimedia, vol. 19, no. 10, pp. 2222–2237, 2017.
[36] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, "Multi-path multi-tier 360-degree video streaming in 5g networks," in Proceedings of the 9th ACM multimedia systems conference, 2018, pp. 162–173.
[37] M. Mahmoud, S. Rizou, A. S. Panayides, N. V. Kantartzis, G. K. Karagiannidis, P. I. Lazaridis, and Z. D. Zaharis, "Optimized tile quality selection in multi-user 360° video streaming," IEEE Open Journal of the Communications Society, 2024.
[38] J. Son, D. Jang, and E.-S. Ryu, "Implementing 360 video tiled streaming system," in Proceedings of the 9th ACM Multimedia Systems Conference, 2018, pp. 521–524.
[39] 3GPP, "Nr; physical channels and modulation (release 16)," 3GPP, Tech. Rep. TS 138.211 V16.2.0 (2020-07), 2020.
[40] D. V. Nguyen, H. Van Trung, H. L. D. Huong, T. T. Huong, N. P. Ngoc, and T. C. Thang, "Scalable 360 video streaming using http/2," in 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP). IEEE, 2019, pp. 1–6.
[41] X. Corbillon, F. De Simone, and G. Simon, "360-degree video head movement dataset," in Proceedings of the 8th ACM on Multimedia Systems Conference, 2017, pp. 199–204.
[42] M. Mahmoud, S. Rizou, A. S. Panayides, N. V. Kantartzis, G. K. Karagiannidis, P. I. Lazaridis, and Z. D. Zaharis, "A survey on optimizing mobile delivery of 360° videos: Edge caching and multicasting," IEEE Access, vol. 11, pp. 68 925–68 942, 2023.
[43] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of shvc: Scalable extensions of the high efficiency video coding standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 20–34, 2015.
[44] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "Qoe-driven rate adaptation heuristic for fair adaptive video streaming," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 12, no. 2, pp. 1–24, 2015.
[45] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, 2015, pp. 325–338.
[46] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in Proceedings of the conference of the ACM special interest group on data communication, 2017, pp. 197–210.
[47] C. Zhou, Y. Ban, Y. Zhao, L. Guo, and B. Yu, "Pdas: Probability-driven adaptive streaming for short video," in Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 7021–7025.
[48] J. De Vriendt, D. De Vleeschauwer, and D. Robinson, "Model for estimating qoe of video delivered using http adaptive streaming," in 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013). IEEE, 2013, pp. 1288–1293.
[49] R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the quality of experience of http video streaming," in 12th IFIP/IEEE international symposium on integrated network management (IM 2011) and workshops. IEEE, 2011, pp. 485–492.
[50] N. V. Hung, T. D. Chien, N. P. Ngoc, and T. H. Truong, "Flexible http-based video adaptive streaming for good qoe during sudden bandwidth drops," EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, vol. 10, no. 2, pp. e3–e3, 2023.
[51] M. U. Younus, "Analysis of the impact of different parameter settings on wireless sensor network lifetime," International Journal of Advanced Computer Science and Applications, vol. 9, no. 3, 2018.