Digital Communications and Networks 10 (2024) 1234–1244

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

Intelligent cache and buffer optimization for mobile VR adaptive transmission in 5G edge computing networks

Junchao Yang [a], Ali Kashif Bashir [b,c,d], Zhiwei Guo [a], Keping Yu [a,e,*], Mohsen Guizani [f]

[a] *Chongqing Key Laboratory of Intelligent Perception and BlockChain Technology, School of Artificial Intelligence, Chongqing Technology and Business University, Chongqing, 400067, China*
[b] *Department of Computing and Mathematics, Manchester Metropolitan University, UK*
[c] *Woxsen School of Business, Woxsen University, Hyderabad 502 345, India*
[d] *Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon*
[e] *Graduate School of Science and Engineering, Hosei University, Tokyo, 184-8584, Japan*
[f] *Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), United Arab Emirates*

ARTICLE INFO

Keywords:
Virtual reality
Adaptive transmission
Edge cache
Buffer management
5G
Mobile edge computing

ABSTRACT

Virtual Reality (VR) is a key industry for the development of the digital economy in the future. Mobile VR has advantages in terms of mobility, lightweight and cost-effectiveness, which has gradually become the mainstream implementation of VR. In this paper, a mobile VR video adaptive transmission mechanism based on intelligent caching and hierarchical buffering strategy in Mobile Edge Computing (MEC)-equipped 5G networks is proposed, aiming at the low latency requirements of mobile VR services and flexible buffer management for VR video adaptive transmission. To support VR content proactive caching and intelligent buffer management, users' behavioral similarity and head movement trajectory are jointly used for viewpoint prediction. The tile-based content is proactively cached in the MEC nodes based on the popularity of the VR content. Second, a hierarchical buffer-based adaptive update algorithm is presented, which jointly considers bandwidth, buffer, and predicted viewpoint status to update the tile chunk in client buffer. Then, according to the decomposition of the problem, the buffer update problem is modeled as an optimization problem, and the corresponding solution algorithms are presented. Finally, the simulation results show that the adaptive caching algorithm based on 5G intelligent edge and hierarchical buffer strategy can improve the user experience in the case of bandwidth fluctuations, and the proposed viewpoint prediction method can significantly improve the accuracy of viewpoint prediction by 15%.

1. Introduction

In recent years, Virtual Reality (VR) application has become a cynosure, that attracts most of the public attention. Applications such as VR sports, VR tourism, VR game, VR conference, VR live broadcast, etc. have been widely popularized [1]. It is predicted that by 2025, the VR application market will reach 30 billion US dollars [2]. Recently, the Internet companies such as Apple, Google, Facebook, and YouTube have also entered the VR market [3,4]. As the most popular application at present, VR uses 360-degree video to build a three-dimensional virtual world in a Head-Mounted Display (HMD), and its features such as immersion, interaction and imagination bring people a new visual interactive experience [5]. However, it poses new challenges to the network [6], as the cost of this new experience is higher transmission bandwidth, higher video bit rate and lower latency requirements.

Compared to traditional video, the data volume of VR video is several times larger due to its panoramic feature. On the one hand, in order to provide an immersive panoramic experience, users need to download the entire panoramic video, but only one-third of the panoramic video (usually the viewing angle of field of view is 90°–120°) is viewed by the user through the HMD [7,8]. The number of pixels that the human eye can see in one degree of viewing angle is called Pixels Per Degree (PPD) [9]. Generally speaking, when the PPD reaches 60 (the limit of the human retina), the clarity of viewing is the highest. Taking a 4K VR video as an example, the horizontal PPD is about 11 (3840/360 = 10.67), which is much lower than 60. Therefore, the current 4K VR video and

* Corresponding author. Chongqing Key Laboratory of Intelligent Perception and BlockChain Technology, School of Artificial Intelligence, Chongqing Technology and Business University, Chongqing, 400067, China.
E-mail addresses: yangjc@ctbu.edu.cn (J. Yang), dr.alikashif.b@ieee.org (A.K. Bashir), zwguo@ctbu.edu.cn (Z. Guo), keping.yu@ieee.org (K. Yu), mguizani@ieee.org (M. Guizani).

https://doi.org/10.1016/j.dcan.2023.07.003
Received 6 September 2022; Received in revised form 16 June 2023; Accepted 2 July 2023
Available online 12 July 2023

standard-definition ordinary video have the same clarity from the viewers' perspective. On the other hand, in order to prevent motion sickness and provide a better Quality of Experience (QoE), the HMD must maintain a high refresh rate (the refresh rate can reach 120 Hz). Similarly, in order to maintain an immersive experience, the VR video frame rate must be kept at a high level (typically 30–100 frames per second (fps)). Therefore, 360-degree video requires a higher resolution (12K) and a higher frame rate (100 fps) than traditional video. As a comparison, 8K traditional video with frame rate of 60 fps uses High Efficiency Video Coding (HEVC) for compression, the code rate is about 100 Mbps [10]. Obviously, the transmission of higher resolution and higher frame rate VR videos is more difficult and challenging compared to traditional video.

Another important factor affecting the VR user experience is latency [11]. In VR applications, VR latency specifically refers to Motion-To-Photon (MTP) latency, which describes time from the start of the user's head movement to the corresponding screen display time. It consists of four main parts: motion sensor delay, network delay, rendering delay and display delay. Research [9] shows that the minimum delay that ensures a high quality VR experience is between 17 and 20 ms. If the MTP delay is greater than 20 ms, it is easy to cause the motion sickness, which leads to a poor VR experience. Although most HMD manufacturers claim that their devices can support a MTP delay of less than 20 ms, unless the network delay can reach several milliseconds (according to the work [12], 80% of the current network delays in China reach more than 90 ms), otherwise sending the whole 360-degree panoramic video to the HMD to reduce the network latency will always be the last resort. Due to the interactive characteristics of VR, the delay requirement is more stringent, so the current mainstream of 4K 360-degree video requires at least hundreds of megabits per second of transmission rate to guarantee the user interaction experience [13]. Thus, to ensure the smooth interactive experience of VR users and the quality of the video in the actual view of users, VR content should be properly prefetched according to the different network conditions of users, and this deserves further research [14,15]. At present, most VR streaming platforms still use the traditional video transmission mechanism, which affects the interactive experience of VR to some extent. Therefore, it is also necessary to study a more suitable transmission mechanism to provide a better VR experience.

The ultimate goal of VR is that the VR user cannot distinguish the boundary between the synthesized virtual world and the real world [16,17]. Therefore, continuously increasing the resolution of VR video to reach the retinal limit of the human eye and eliminating the limitation of wired connections are two important steps toward this ultimate goal. Undoubtedly, mobile VR has extremely high requirements for large bandwidth, ultra-reliability, and low latency: under the low latency constraint, gigabits of data per second must be transmitted to the user terminal [18]. As we all know, low latency and high reliability are two contradictory requirements. Ultra-reliability requires to be allocated to ensure the transmission success rate, while this will increase the delay of other users. Obviously, to realize mobile VR, intelligent network design is required to meet the requirements of reliability, latency and seamless support for different network scenarios [11].

5G has the characteristics of bandwidth-intensive, low latency and wide connectivity, which provides the infrastructure for the implementation of mobile VR [19,20]. Among the many application scenarios of 5G, mobile VR is considered to be the first killer application of 5G, and has received extensive attention from academia and industry. Mobile Edge Computing (MEC) is an important paradigm of 5G. Its core idea is to bring the services, content and resources closer to the end-user through software-defined networking and network function virtualization technology [21,22]. MEC offers a possible solution to the problems faced by mobile VR transmission [23,24]. The main advantages of MEC are: 1) Since the computational and storage resources of MEC are closer to the users, the communication delay can be significantly reduced. 2) Intensive computing tasks of users can be offloaded to MEC nodes to solve the problem of insufficient computation of mobile terminals. 3) The content can be cached on the MEC node through an efficient caching strategy to improve the utilization of storage resources.

Proactive caching of VR content in MEC will improve the VR user's QoE in terms of latency to some extent, but buffer management in the terminal is another key factor that directly affects the visual perception [13]. Therefore, caching and buffer management should be optimized together to provide a better experience for VR users. Therefore, in response to the current problems of content caching and user terminal buffer management in the 5G edge computing for mobile VR, this paper studies the edge caching and buffer optimization problem. The main contributions of this paper are summarized as follows.

- A mobile VR adaptive transmission mechanism based on 5G intelligent edge is proposed. VR content is proactively cached to the MEC nodes according to the popularity of VR tiles, and a buffer adaptive update algorithm based on viewpoint prediction is implemented to improve user experience.
- In order to improve the viewpoint prediction accuracy, the viewpoint prediction method based on the user's historical trajectory and the motion similarity between users is proposed, thereby improving the cache hit rate of edge nodes and optimizing the user buffer management.
- An adaptive update algorithm for buffer management that jointly considers user bandwidth, buffer status, and predicted viewpoint is proposed based on a hierarchical buffer strategy. The problem formulation and the corresponding algorithm are presented.
- Finally, the simulation shows that the proactive caching scheme cooperates with the proposed adaptive update algorithm can guarantee better performance(QoE) with the bandwidth fluctuation, meanwhile reduce the latency of VR users.

The rest of this paper is organized as follows. In Section 2, the related work is outlined. In Section 3, the system model is described. In Section 4, the mobile VR adaptive transmission mechanism is proposed. In Section 5, the performance evaluation and discussion are shown. Finally, in Section 6, the paper is summarized.

2. Related work

MEC has brought hope to the implementation of mobile VR, but it has not completely solved all the challenges of mobile VR content caching in the MEC server and buffer management in the user terminal still need to be carefully designed to achieve better QoE. On the one hand, the viewpoint prediction based on the trajectory of the user's head movement significantly reduces the accuracy of the prediction over time, which increases the difficulty of proactive content caching. On the other hand, most VR streaming applications still use the traditional video buffer management mechanism, which will degrade the interactive experience of VR to some extent. Therefore, more accurate viewpoint prediction methods and intelligent buffer management are needed to deal with the aforementioned problems. The user viewpoint prediction methods based on motion [15] and content [25] are two mainstream methods that are widely used. The motion-based viewpoint prediction is mainly based on the user's historical trajectory to predict the future viewpoint, and its accuracy drops sharply along with the prediction time. The content-based viewpoint prediction is based on the video structure, saliency and other characteristics of the content to predict the user's viewpoint. However, the effect mechanism on the user's viewpoint is not very clear. Recently, the work [26] tried to use the K-Nearest-Neighbors (KNN) method to predict the user's viewpoint based on the behavioral similarity between users, and this method achieved good prediction results.

Caching in MEC will improve the VR user's QoE to some extent in terms of latency, but the buffer management in the terminal is another key factor that directly affects the visual perception. Therefore, buffer management still needs to be carefully designed. In terms of buffering mechanism, the work [23,24] realized that the solution of transmitting

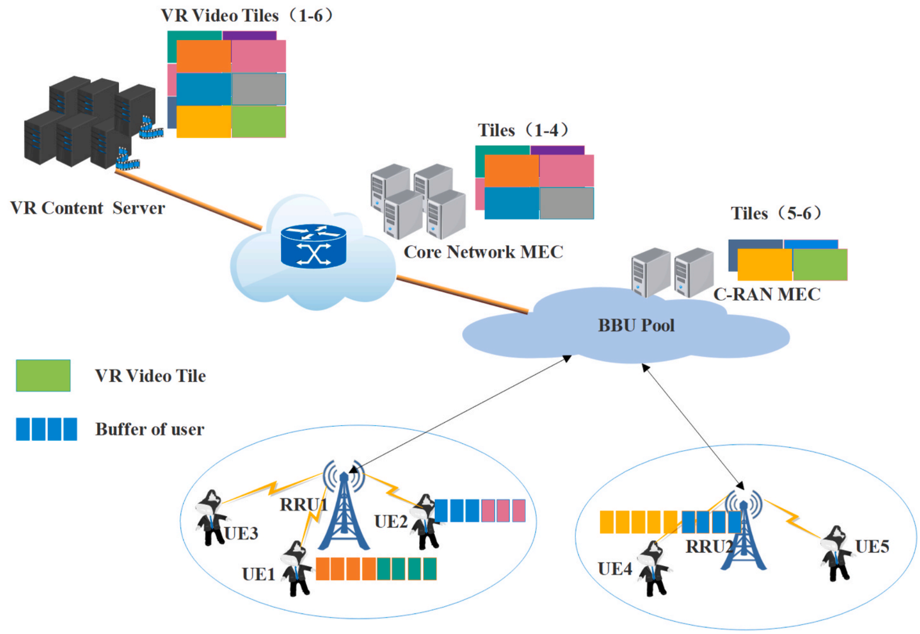


Fig. 1. The system model.

the higher video quality from the current user's point of view can save bandwidth to some extent, but it increases the spatial dimension to buffer future videos. Once rebuffering occurs, video freezing is likely to occur, affecting the user's QoE. In Ref. [27], an adaptive 360-degree video transmission scheme based on scalable coding was proposed, the main idea is to use the characteristics of the base layer and the enhancement layer of scalable coding for adaptive buffering. In scalable coding, the base layer of the video is required, which can be buffered with a longer time granularity by ignoring the user's viewpoint. Since the user's viewpoint changes quickly, the enhancement layer can buffer a shorter time granularity to improve the user QoE. [28] also uses the idea of scalable coding, the user's viewpoint strategy based on scalable coding can solve the rebuffering problem, but its coding complexity is relatively high. In Ref. [29], a hierarchical buffering strategy was proposed for adaptive transmission of VR video, but it lacked optimization of the bitrate selection in the buffer management.

However, according to our research, neither proactive content caching at the edge nor buffer content optimization at the client side can guarantee a high quality of user experience under current network conditions. Therefore, joint optimization of caching and buffer management is probably the only way to meet the demanding latency and bandwidth requirements of VR, and this is the main motivation of this paper.

3. System model

The system model of this paper is shown as Fig. 1, which includes four components: content server, access network, edge node and user terminal. In the following part, the function of each component is described. The content server is responsible for storing video content and responding to user requests. The access network uses a typical 5G Centralized Radio Access Network (C-RAN) to provide transmission services to users through a Remote Radio Unit (RRU). In the C-RAN MEC, edge computing servers are deployed in the Base Band Unit (BBU) to provide storage and computing resources. In this paper, a multi-level edge computing architecture is considered, where network MEC is deployed in other nodes of the core network and has the same function as the C-RAN MEC. This paper takes mobile VR headsets as the main research objects, which provide display, interaction, communication and simple computation functions. An adaptive content update module is deployed for the mobile VR terminal, which makes intelligent buffer content management update decisions based on bandwidth status, user's predicted viewpoint, and current buffer status. Finally, the user terminal performs the rendering of the corresponding viewpoint according to the head movement. In addition, the content server records the user's viewpoint trajectory, performs viewpoint prediction based on the historical motion trajectory and the viewpoint trajectory of similar users, and feeds back the result to the viewpoint prediction for the adaptive decision module of the user terminal in time.

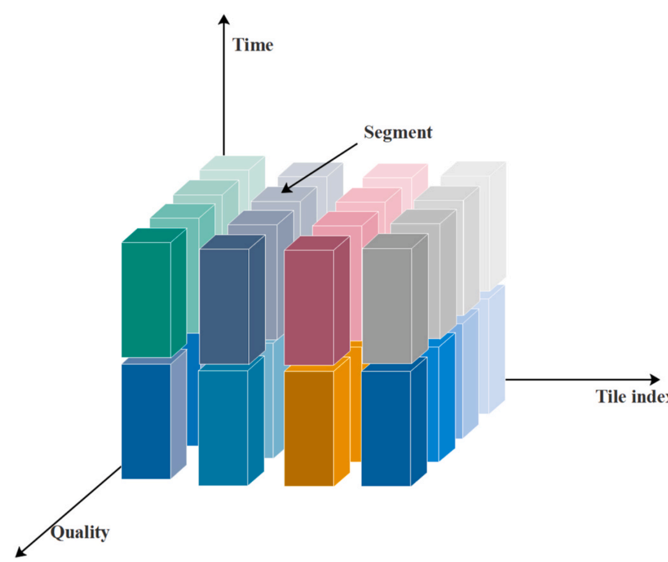


Fig. 2. Tile-based VR video coding.

In addition, we adapt tile based VR encoding for content delivery in this paper. Fig. 2 shows an example of tile-based VR video encoding based on Equi-Rectangular Projection (ERP) [30,31]. The VR video is divided into N segments the temporal dimension, and the encoder divides each video segment in the spatial dimension independently by encoding as tiles, and each tile is encoded into M bitrate versions. In this paper, a particular code rate version of the nth segment is called a tile chunk. Each tile produces multiple bitrate versions, i.e., $R_t \in \{R_t^l, \dots R_t^Q\}$, where R_t^l and R_t^Q represent the minimum and the maximum bitrates, respectively. The segment temporal dimension is denoted as $n \in N$, and

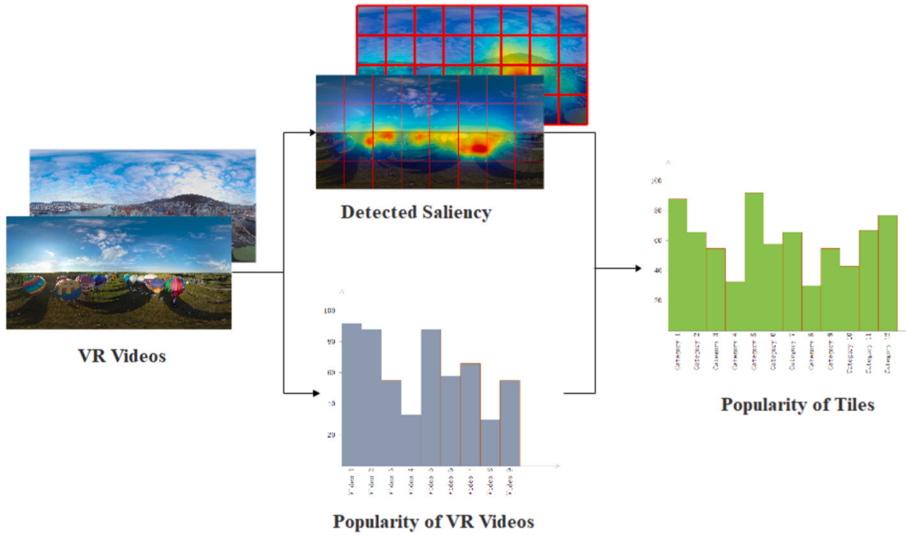


Fig. 3. Video popularity and user interest based tile ranking.

the bitrate of the t-th tile of the n-th segment is denoted as $R_{n,t}$.

4. Methodology

The accuracy of user viewpoint prediction is an important factor affecting the performance of the mobile VR. On the one hand, it improves the cache hit rate of edge nodes for the edge caching. On the other hand, the optimization of user-side buffer management is also based on viewpoint prediction. In order to improve the accuracy of viewpoint prediction, this paper proposes a cross user viewpoint prediction method, which requires the viewpoint trajectories of similar users and the trajectories of user motion together. The viewpoint prediction is performed in the DASH server, and then the viewpoint prediction result is fed back to the user terminal's adaptive module in time. The adaptive module makes a buffer adaptive update decision based on the bandwidth status, the predicted viewpoint of the user, the status of the current buffer, and the requested VR video chunks are proactively buffered in the buffer of the user terminal. Finally, the user terminal renders the video in the buffer according to the user's current viewpoint. In the following part, the proposed viewpoint prediction method and buffer adaptive update algorithm are elaborated.

4.1. Edge caching of VR content based on tile popularity

The mobile network integrates the MEC, which provides computing and storage functions and brings VR content closer to the users [32,33]. If the video content requested by the user is cached at the edge node within its acceptable service range, the user can obtain the video content directly from the edge node, without going through the core network or obtaining the content from the remote video server, and the service delay can be effectively reduced. On the one hand, the storage space of edge nodes is limited, and it is extremely difficult to cache all VR video content on edge nodes [34]. On the other hand, the service delay for users can be significantly reduced, and the operation cost and energy consumption of edge nodes can be reduced by improving the content cache hit rate of edge nodes.

According to the characteristics of the VR content request based on the user's point of view, this paper adopts the cache based on the interest degree and content popularity prediction with tile granularity, as shown in Fig. 3. Our previous work has been confirmed that the user's view is concentrated in a region with strong video saliency [22], and the saliency detection could be applied to obtain the τ segment of VR video n of tile m's saliency value $P_{n,\tau,m}$. And the popularity of VR video n in the edge node k P_n^k is defined as Equation (1). Finally, the saliency of the tile (i.e., the degree of interest) and the popularity of the VR video are used to predict $P_{n,\tau,m}^k$ the popularity of the tile, which is defined as the weighted sum function, and shown as Equation (2). In this paper, we adopt the edge caching based on tile granularity to improve cache hit rate and storage efficiency.

$$p_n^k = \frac{\left(\sum_n^N \left(\gamma_{n,k}^{-\alpha}\right)\right)^{-1}}{\gamma_{n,k}^{-\alpha}} \tag{1}$$

$$p_{n,\tau,m}^k = \varphi \cdot p_{n,\tau,m} \times \lambda \cdot p_n^k \tag{2}$$

Based on edge caching, deploying edge computing and edge caching in the core network improves user experience quality while reducing transmission delay. In addition, in-network caching has been proven in NDN network to greatly reduce. In this paper, the VR video content are cached at the core network edge and C-RAN edge cache, is referred to as the multi-level cache, which is shown in Fig. 1. Multi-level caching has the following advantages [35,36]: 1) Caching tile-granular VR video content to edge nodes to reduce service delay. 2) Edge caching based on viewpoint prediction greatly improves the caching efficiency of edge nodes. 3) Caching based on tile granularity improves the edge nodes to some extent and reduces edge storage cost. 4) Multi-level caching further improves the cache hit rate.

4.2. Cross user viewpoint prediction

Tile-based adaptive transmission is an efficient way to transmit VR video. However, because VR applications are sensitive to delay, and it is impossible to deliver high-quality VR video to the user in time. The user needs a large buffer to store the precached video content for a smooth viewing experience. Therefore, if the user's viewpoint can be predicted in advance, then the quality of the viewpoint part of the tile can be enhanced in advance, it can save bandwidth and improve the transmission to a great extent. However, the accuracy of the viewpoint prediction based on the trajectory of the user's head movement decreases significantly with time, so it is necessary to explore new viewpoint prediction methods.

In the work [23], they use K viewpoints of other users that are closest to the predicted viewpoint to correct the value of user motion based prediction, but they ignored the similarity of behavior between users. In their case, using the closest K user viewpoints to correct motion-based prediction results is very likely to lead to larger deviations in the final prediction results. On the other hand, the studies in Ref. [28] have

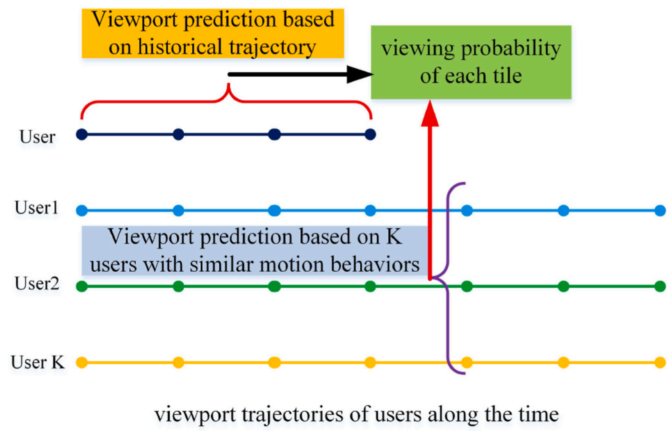


Fig. 4. Illustration of cross user viewpoint prediction.

pointed out that when watching the same VR video, the viewing behavior of users in the virtual environment is similar to a certain extent. Inspired by their work, this paper attempts to use the similarity of motion between users to improve the prediction accuracy of the user's viewpoint. Fig. 4 shows an illustration of the proposed viewpoint prediction method. First, the user's viewpoint is predicted based on the user's historical trajectory, then the predicted user's viewpoint is corrected by using the similarity of motion between users. The specific method is described below.

First, we defined the motion similarity of two users as the ratio of the number of tiles where the viewpoints of two users overlap in a period of time, as shown in Equation (3).

$$Sim\{v,j\} = \sum_{n_\delta=0}^{D} \frac{2\Omega_{n_\delta}\{v,j\}}{\Theta_{n_\delta}\{v\} + \Theta_{n_\delta}\{j\}} \tag{3}$$

Where, n_δ is the sampling index of the user's viewpoint in nth segment. D is the time window of the similarity measurement. $\Omega_{n_\delta}\{v,j\}$ and $\Theta_{n_\delta}\{v\}$ represent the number of tiles where the two user's viewpoints overlap at n_δ time and the number of tiles within the user's viewpoint, respectively.

It is assumed that the motion trajectories of all users are recorded by the VR content server. Based on the similarity measurement between users, the viewpoint prediction module selects K most similar users to the predicted user viewpoint, and uses the motion trajectories of the K most similar users to correct the predicted user viewpoint. Consider adopting a comprehensive weight voting mechanism based on motion trajectory prediction and K most similar user viewpoints. Among them, the Linear Regression (LR) method based on motion trajectory prediction is used. A linear regression model for predicting the user's viewpoint in a video chunks is shown in Equation (4). Among them, $O_{LR}(\alpha,\beta,\gamma)$ represents the viewpoint predicted by the linear regression method, α, β and γ are the attitude angles of the X, Y, and Z axes, also called Euler angles; ω_{LR} is the regression coefficients of the linear regression; Equation (4) is for converting the Euler angles of the viewpoint into 3D coordinates.

$$\begin{cases} \alpha_{LR}(\mathrm{t}_0+\ \Gamma) = \omega_\alpha\Gamma + \alpha(\mathrm{t}_0) \\ \beta_{LR}(\mathrm{t}_0+\ \Gamma) = \omega_\beta\Gamma + \beta(\mathrm{t}_0) \\ \gamma_{LR}(\mathrm{t}_0+\ \Gamma) = \omega_\gamma\Gamma + \gamma(\mathrm{t}_0) \end{cases} \tag{4}$$

$$\begin{cases} x = \sin(\alpha)\cos(\beta) \\ y = \sin(\beta) \\ z = \cos(\alpha)\cos(\beta) \end{cases} \tag{5}$$

$$\xi_t = W_{LR}\cdot F_t(O_{LR}) + \sum_{k}^{K} W_{SM}\cdot F_t(O_{SM}^k) \tag{6}$$

Tile's vote ξ_t can be expressed as Equation (6), where W_{LR} and W_{SM} represent the weight of the predicted viewpoint based on the linear regression method and the weight of the similar user viewpoint, respectively. $F(O)$ represents the range covered by the viewpoint, which is a T-dimensional vector, and T is the index of the tile in the raster scan order. $F_t(O_{LR})$ and $F_t(O_{SM}^k)$ represent t-th tile are within the viewpoint range according to the linear regression and the K most similar users methods, respectively. $F(O) = 1$ means that the t-th tile is within the viewpoint range, and is equal to 0 if the tile is outside the predicted viewpoint range. O_{SM}^k means the viewpoint of the k-th most similar users. In addition, since the accuracy of the linear regression prediction method based on historical trajectory decreases with time, the weight of W_{LR} the linear regression method is inversely proportional to time, using the form of Equation (7), and W_{SM} is set as 1 since the summation of $F_t(O_{SM}^k)$ denotes the ratio of the tile in the viewpoint range for all the K similar user.

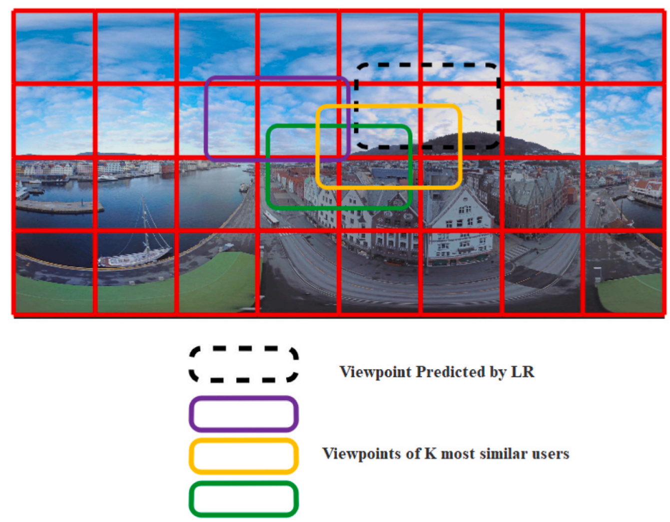


Fig. 5. An example of voting mechanism for cross user viewpoint prediction.

$$W_{LR} = \frac{1}{\Gamma} \tag{7}$$

Fig. 5 shows an illustration of the weight-based voting mechanism. The dark dashed box is the viewpoint based on LR prediction, and the other blue, green, and yellow boxes are the historical viewpoints of similar users. According to our proposed mechanism, the weights of each tile are calculated by Equation (6). Obviously, these weights are the predicted probability that the user's viewpoint is focused on these tiles.

Algorithm 1 *P*4 solving algorithm

INPUT: The bitrate of tile in the downloaded video chunks $R_{n,t}^{do}$, The bitrate of the existing tiles in the buffer $R_{n,t}$, minimum and maximum tile bitrate R_t^l, R_t^Q

OUTPUT: The index of the updated tile $x_{n,t*}$ and the bitrate of the updated tile $R_{n,t*}^{re}$

for all tile in $[0, B_{Cur}]$ **do**
  for all tile code rate versions **do**
    Calculate all tile utility over cost ratio according to Equation (15)
    Arrange in descending order
  end for
end for
Initialization sum = 0
While $sum \le R(t_c) - \sum_t^T R_{n,t}^{do}$ **do**
  Mark the tile with maximum utility over cost ratio in the assign queue and assign it, i.e.$x_{n,t*} = 1$
  Update $sum = sum + \sum_{n\in[0,B_{Cur}],t*\in T} x_{n,t*}R_{n,t*}^{re}$
end while

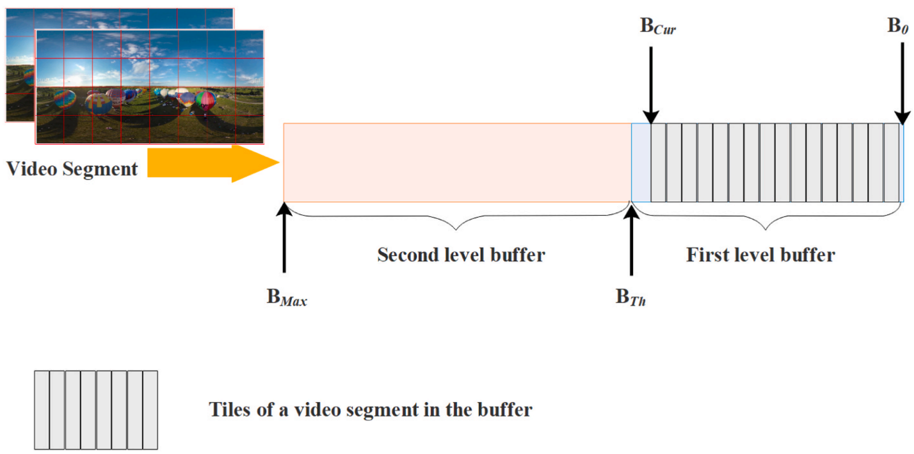


Fig. 6. The illustration of hierarchical buffer.

4.3. Adaptive update algorithm based on hierarchical buffer

Pre-caching VR content at edge nodes significantly reduces network transmission delay. However, the time-varying characteristics of mobile network bandwidth and the randomness of users viewing VR content determine the need for content caching at user terminals. For delay-sensitive VR applications, user terminal caching can provide a smooth user experience, but if the user's bandwidth condition is good, if the low-quality video chunks are buffered for a long time, the buffer overflow will occur. If high-quality video is cached for a short period of time, the video chunks cannot be downloaded in time, resulting in a freezing phenomenon, both of which will degrade the user experience. Therefore, based on viewpoint prediction, we propose an adaptive transmission algorithm based on hierarchical buffering, which comprehensively considers the user's bandwidth status, the viewpoint prediction and the state of the buffer region for adaptive transmission. Specifically, the bandwidth status of the user adopts a linear regression method that uses the status of the previous time period to predict the bandwidth of the next time period. The linear regression model is shown in Equation (8), where, $BW(t_0)$ represents the bandwidth status in the previous time period, μ_{LR} represents the regression coefficient of linear regression, and Γ is the time length of a video chunk. We used locally weighted LR for bandwidth prediction. For the locally weighted linear regression algorithm, the whole training data is required for each prediction (each prediction yields a different parameter μ_{LR}).

$$BW(t_0+\Gamma)=\mu_{LR}\Gamma+BW(t_0) \tag{8}$$

In this paper, we assume that the content server records the trajectory of the user's viewpoint, and performs viewpoint prediction based on the historical trajectory and the viewpoint trajectory of similar users, and then feeds back the result of the viewpoint prediction to the adaptive module of the user terminal in real-time.

Algorithm 2 Adaptive algorithm

INPUT: Maximum buffer B_{Max}, buffer region threshold B_{Th}, predicted bandwidth $BW(t_c)$, current buffer state B_{Cur}, tile bitrate of the downloaded video chunks $R^{do}_{n,t}$
OUTPUT: Update tile index $x_{n,t*}$, and bitrate of update tile $R^{re}_{n,t*}$
1 : **if** $B_{Cur} \le B_{Th}$
2 : Only download video content in $[0, B_{Th}]$
3 : According to P1 solution x_{n,t^*} and R^{re}_{n,t^*}, update tile content in $[0, B_{Cur}]$
4 : According to P1 solution $R^{do}_{n,t}$, download new content in $[B_{Cur}, B_{Th}]$
5 : **else when** $B_{Cur} \ge B_{Th}$
6 : Based on P2 solution x_{n,t^*} and x_{n,t^*}, update tile content in $[0, B_{Th}]$
7 : Based on P2 solution $R^{do}_{n,t}$, download new content in $[B_{Cur}, B_{Max}]$
8 : **end if**

The hierarchical buffer strategy proposed in Ref. [37], which divides the entire buffer into two regions, namely $[0, B_{Th}]$ and $[B_{Th}, B_{Max}]$, where B_{Th} and B_{Max} represent the hierarchical threshold and maximum buffer value, respectively. The hierarchical buffer is shown in Fig. 6, where $[0, B_{Th}]$ is the first-level buffer region. If the user's current buffer state B_{Cur} is less than B_{Th} that, it means that the content in the buffer is being consumed quickly. If new video chunks are not downloaded in time, video playback will be stuck, it is necessary to request more video chunks in time to meet the needs of users. Therefore, the strategy adopted is to update the $[0, B_{Cur}]$ part of the tile based on the result of the user's viewpoint prediction, that is, the tile that brings greater utility value gain to transmit a higher-quality version. For part of the $[B_{Cur}, B_{Th}]$, by downloading the video to reach the buffer region threshold B_{Th}.

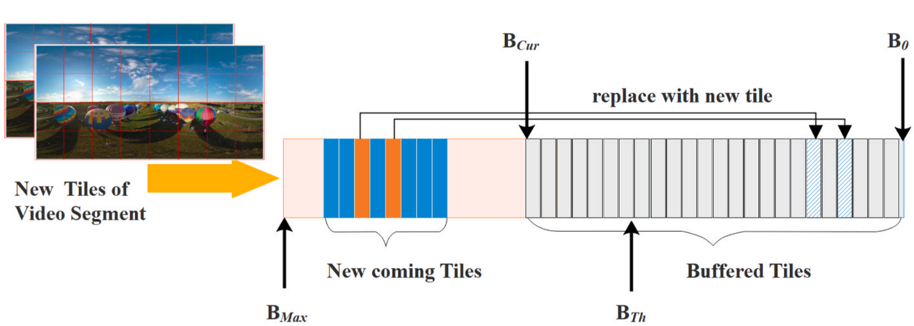


Fig. 7. Content update of hierarchical buffer.

For the downloading task of this part, the optimization of the utility value in the whole buffer region is achieved by the selection of the tile code rate in the video chunks. Content update of hierarchical buffer is shown in Fig. 7, and the mathematical description of the problem is formulated as follows:

$$\begin{aligned} P1: &\max_{x_{n,t^*}, R^{re}_{n,t^*}, R^{do}_{n,t}} \sum_{n\in[0,B_{Cur}]} \sum_{t}^{T} x_{n,t^*}\left(U^{re}_{n,t^*} - U_{n,t}\right)p_{n,t} + \sum_{n\in[B_{Cur},B_{Th}]} \sum_{t}^{T} U^{do}_{n,t}p_{n,t} \\ &\text{s.t.} \sum_{t}^{T} x_{n,t^*}R^{re}_{n,t^*} + \sum_{t}^{T} R^{do}_{n,t} \le R(t_c), \forall n \in [0, B_{Th}], \forall t, \\ &\quad t^* \in T (C1) \\ &R^{re}_{n,t^*}, R^{do}_{n,t} \in \{R^1_t, ..R^Q_t\}, \forall n \in [0, B_{Th}], \forall t, t^* \in T (C2) \\ &x_{n,t^*} \in \{0,1\}, \forall n \in [0, B_{Th}], \forall t, t^* \in T (C3) \end{aligned} \tag{9}$$

Among them, $p_{n,t}$ represents the probability that the t tile of the n video chunks is viewed by the user. x_{n,t^*} indicates whether the t tile of the n video chunks is updated with a higher quality version, and R^{re}_{n,t^*} indicates the updated quality version of the t tile of the n video chunks. $R^{do}_{n,t}$ represents the downloaded quality version of the t tile of the n video chunks. U^{re}_{n,t^*}, $U_{n,t}$ and $U^{do}_{n,t}$ represent the utility value of the new quality version of the t tile of the n video chunks, the utility value of the t tile of the n video chunks in the buffer region and the utility value of the new video chunks that needs to be downloaded, respectively. $R(t_c)$ indicates the downloads data rate of the user.

In this paper, the utility function is used to measure the quality of VR video transmission. The utility value of the tile chunks can be expressed as Equation (10), where $R_{n,t}$ and $R^Q_{n,t}$ represent the bitrate of the t-th tile chunks and the maximum bitrate of the t-th tile chunks of the n-th video chunks, respectively. α and β are the coefficients of the utility function. The utility function is a convex function of the video bitrate, and the first-order derivative of the function decreases as the video bitrate increases. These characteristics of the utility function can well model the user's viewing experience. For tile-based VR videos, the utility value of a tile describes its contribution to the user's overall viewing experience at a given bitrate.

$$U_{n,t} = \begin{cases} \alpha \log\left(\beta \dfrac{R_{n,t}}{R^Q_{n,t}}\right), R^Q_{n,t} > 0 \\ 0, R_{n,t} = 0 \end{cases} \tag{10}$$

If the current buffer status of the user B_{Cur} is greater than B_{Th}, it means that there are enough video chunks in the buffer for the user to play, so the tile of the [0,B_{Th}] part is updated; the video chunks of the [B_{Cur}, B_{Max}] part are downloaded, because the prediction accuracy of the user's viewpoint is low when downloading the video chunks of this part, the tile selection will have a large deviation, and the uniform lowest bitrate version of the video chunks in the [B_{Cur}, B_{Max}] part is downloaded. Similar to problem P1, the mathematical description of the problem is as follows.

$$\begin{aligned} P2: &\max_{x_{n,t^*}, R^{re}_{n,t^*}, R^{do}_{n,t}} \sum_{n\in[0,B_{Th}]} \sum_{t}^{T} x_{n,t^*}\left(U^{re}_{n,t^*} - U_{n,t}\right)p_{n,t} + \sum_{n\in[B_{Cur},B_{Max}]} \sum_{t}^{T} U^{do}_{n,t} \\ &\text{s.t.} \sum_{t}^{T} x_{n,t^*}R^{re}_{n,t^*} + \sum_{t}^{T} R^{do}_{n,t} \le R(t_c), \forall n \in [0, B_{Max}], \forall t, \\ &\quad t^* \in T (C1) \\ &R^{re}_{n,t^*}, R^{do}_{n,t} \in \{R^1_t, ..R^Q_t\}, \forall n \in [0, B_{Max}], \forall t, t^* \in T (C2) \\ &x_{n,t^*} \in \{0,1\}, \forall n \in [0, B_{Max}], \forall t, t^* \in T (C3) \end{aligned} \tag{11}$$

On the one hand, when the content of the buffer region is about to be consumed, it is necessary to adopt a conservative strategy to update the buffer region's tile and download new video chunks. On the other hand, when the buffer region is about to reach the maximum buffer value, a more aggressive approach is needed. This strategy updates the tiles in the buffer region and downloads new video chunks. Therefore, the requested download rate $R(t_c)$ must be determined by combining the current buffer state and the predicted bandwidth. The download rate $R(t_c)$ is determined by Equation (12) [37], where κ is the proportional coefficient.

$$R(t_c) = \kappa^{B_{Max}-B_{Cur}} \cdot BW(t_c) \tag{12}$$

Note that the problems P1 and P2 are both nonlinear mixed integer optimization problems. Based on the above analysis, it can be concluded that different download strategies B_{Cur} need to be adopted depending on the current buffer state. Next, the corresponding solution algorithms for $P1$ and $P2$ based on different download strategies are given.

4.3.1. Solution of problem P1

If the user's current buffer state B_{Cur} is less than B_{Th}, a conservative strategy is needed to update the tiles in the buffer region and download new video chunks to ensure the user's experience. Enough video chunks must be kept in the buffer region to accommodate changes in network bandwidth. Therefore, downloading new video chunks has a higher priority than updating the buffer tile task. The following strategy can be adopted: first download the [B_{Cur}, B_{Th}] video chunks according to the weight of the viewpoint prediction, and then update the [0, B_{Cur}] part of the tile according to the weight of the viewpoint prediction when bandwidth resources are left. Therefore, the problem P1 can be decomposed into two sub-problems P3 and P4 for solution, where P3 is the optimization problem of downloading [B_{Cur}, B_{Th}] part of the video chunks based on the weight of the viewpoint prediction, and introduces the proportion of the total bandwidth ρ of the downloaded video chunks, where $0 \le \rho \le 1$. In this paper, ρ is set to 0.7 based on experience; and $P4$ is the optimization problem of updating the [0, B_{Cur}] part of the tile based on the weight of viewpoint prediction.

$$\begin{aligned} &P3: \max_{R^{do}_{n,t}} \sum_{n\in[0,B_{Cur},B_{Th}]} \sum_{t}^{T} U^{do}_{n,t}p_{n,t} \\ &\text{s.t.} \sum_{t}^{T} R^{do}_{n,t} \le \rho R(t_c), \forall n \in [0, B_{Th}], \forall t \in T (C1) \\ &R^{do}_{n,t} \in \{R^1_t, ..R^Q_t\}, \forall n \in [0, B_{Th}], \forall t, t^* \in T (C2) \end{aligned} \tag{13}$$

Obviously, after relaxing condition $C2$ to $R^1_t \le R^{do}_{n,t} \le R^Q_t$, the optimization problem P3 is a convex optimization problem, and the optimal solution can be obtained by using a heuristic algorithm or by using convex optimization tools after the condition is relaxed.

$$\begin{aligned} &P4: \max_{x_{n,t^*}, R^{re}_{n,t^*}} \sum_{n\in[0,B_{Cur}]} \sum_{t}^{T} x_{n,t^*}\left(U^{re}_{n,t^*} - U_{n,t}\right)p_{n,t} \\ &\text{s.t.} \sum_{t}^{T} x_{n,t^*}R^{re}_{n,t^*} \le R(t_c) - \sum_{t}^{T} R^{do}_{n,t}, \forall n \in [0, B_{Th}], \forall t^* \in T (C1) \\ &R^{re}_{n,t^*} \in \{R^1_t, ..R^Q_t\}, \forall n \in [0, B_{Th}], \forall t^* \in T (C2) \\ &x_{n,t^*} \in \{0,1\}, \forall n \in [0, B_{Th}], \forall t^* \in T (C3) \end{aligned} \tag{14}$$

Considering problem P4, the optimization objective maximizes the utility value gain by updating the [0, B_{Cur}] part of the tile. Based on the solution strategies for such problems in the works [17,34], the utility-over-cost ratio function is defined to reflect the ratio of utility value gain to bandwidth consumption, which is shown as follows.

$$C_{n,t} = p_{n,t} \frac{U^{re}_{n,t^*} - U_{n,t}}{R^{re}_{n,t^*}} \tag{15}$$

By calculating the value of the utility to cost ratio of all the tiles in the [0, B_{Cur}] part, and then sorting them, and sequentially selecting the tiles with the largest utility-over-cost value to update until the available bandwidth resources are allocated. The specific algorithm steps to solve the P4 problem are shown in Algorithm 1.

4.3.2. Solution of problem P2

If the user's current buffer state B_{Cur} is greater than B_{Th}, it means that there are enough video chunks in the buffer for users to play. The buffer overflow can occur if a conservative strategy is used. An aggressive strategy is required to update the tiles in the buffer and download new video chunks. On the other hand, a fraction of video chunks larger than B_{Th}, the accuracy of the viewpoint prediction decreases, and the rate allocation based on the viewpoint prediction weight causes a waste of resources. Based on the above considerations, an aggressive bitrate update strategy is adopted for the [0,B_{Th}] part of the tiles, and the [B_{Cur}, B_{Max}] part of the video chunks is downloaded using the unified lowest bitrate version. In the case where the [B_{Cur}, B_{Max}] part of the video chunks rate has been determined, problem $P2$ can be rewritten as follows.

$$\begin{aligned}
&\text{P5}: \underset{x_{n,t^*},R^{re}_{n,t^*}}{\text{Max}} \sum_{n\in[0,B_{Th}]} \sum_{t}^{T} x_{n,t^*}\left(U^{re}_{n,t^*} - U_{n,t}\right)p_{n,t} \\
&\text{s.t.} \sum_{t}^{T} x_{n,t^*}R^{re}_{n,t^*} \le R(t_c) - \sum_{t}^{T} R^{do}_{n,t}, \forall n \in [0, B_{\text{Max}}], \\
&\qquad \forall t^* \in T(C1) \\
&R^{re}_{n,t^*} \in \left\{R^1_t, ..R^Q_t\right\}, \forall n \in [0, B_{\text{Max}}], \forall t^* \in T(C2) \\
&x_{n,t^*} \in \{0,1\}, \forall n \in [0, B_{\text{Max}}], \forall t^* \in T(C3)
\end{aligned} \tag{16}$$

Note that problem P5 is similar to P4, so algorithm 1 can be used to solve it, and the specific process will not be repeated here. At this point, the tile rate selection problem for adaptive buffer update is solved.

Finally, based on the solution of the P1 and P2 problems, the user terminal adapts an adaptive VR content request according to the bandwidth situation, the state of the buffer region and the viewpoint prediction. The proposed adaptive algorithm is shown in Algorithm 2. When the user's current buffer status B_{Cur} is less than B_{Th}, the [0, B_{Cur}] part of the tile is updated based on the result of viewpoint prediction, the tile that brings greater utility value gains transmits a higher-quality version; at the same time, for the [B_{Cur}, B_{Th}] part, by downloading this part of the video chunks to reach the threshold B_{Th} of the buffer region in time. If the user's current buffer status B_{Cur} is greater than B_{Th}, update the [0, B_{Th}] part of the tile; at the same time, download the [B_{Cur}, B_{Max}] part of the video chunks with the unified lowest bitrate version.

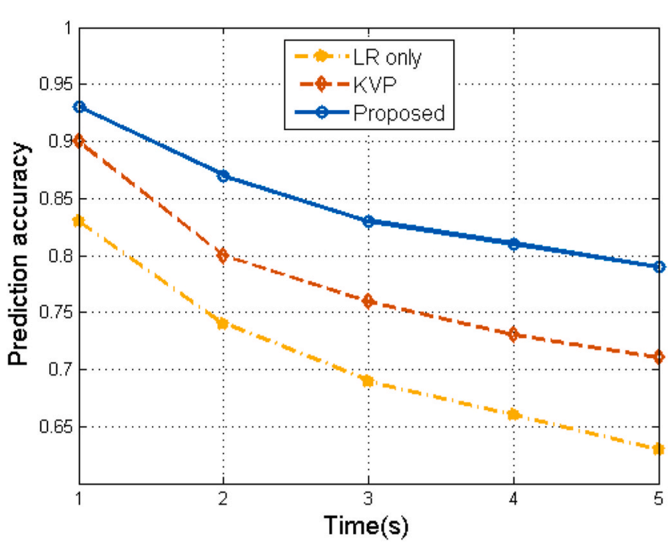


Fig. 8. The accuracy of viewpoint prediction with different methods.

5. Simulation verification and performance analysis

5.1. Simulation settings

The video source of 4K resolution VR video (ie, Freestyle Skiing) in the dataset provided by work [27] is adopted for the simulation, where the number of tiles is 4 × 8, and the open source HEVC encoder Kvazaar is used for tile-based compression encoding. Each tile generates 10 bitrate versions, which are 0.1, 0.3, 0.5, 0.7, 0.9, 1, 1.2, 1.5, 1.7, 2.0 Mbps. In addition, the work [38] provides the viewpoint trajectory of 48 users while watching the VR video. In the simulation, one user is selected as the predicted user. The first half of the viewpoint trajectory is used for prediction, and the second half is used to evaluate the performance of viewpoint prediction. For a specific user, K similar viewpoint motion users are first selected based on the similarity measure between users; at the same time, based on the historical viewpoint trajectory, linear regression is applied to predict the user viewpoint motion in the next video chunks. Then, by integrating the viewpoint trajectories of K users and the viewpoints predicted by the linear regression method, the probability of the user's viewpoint falling on each tile is obtained based on the voting mechanism proposed in this paper. Finally, the real user viewpoint trajectory is used in the second part to objectively evaluate the user QoE by measuring the bitrate of the tile in the user's real viewpoint.

The user bandwidth data set is provided in the work [39], the bandwidth changes with the movement of the user, and the peak bandwidth can reach 95 Mbps. Since the length of the dataset varies from 166 s to 758 s, two typical bandwidth changes and fluctuations are selected, namely pedestrian bandwidth trace 1 (smaller bandwidth change) and pedestrian bandwidth trace 5 (larger bandwidth change), and take the first 50 s of records for simulation, respectively.

The maximum value of the user buffer region is set to 5 s, i.e., $B_{\text{Max}} = 5$, and the hierarchical buffer threshold is set to 2 s, i.e., $B_{Th} = 2$. In addition, the Traditional Buffer (TB) strategy [40] and the Two-Tier Buffer (TTB) strategy [13] are two typical buffer strategies that are compared with the proposed adaptive Hierarchical Buffer (HB) strategy. Among them, the traditional buffer strategy buffer size is set to 5 s; the two-layer buffer strategy, the buffer threshold is set to 2 s to buffer the enhancement layer video, 3 s to buffer the base layer video.

5.2. Performance analysis

First, the cross-user viewpoint prediction method proposed in this paper is compared with other methods. Among them, the traditional linear regression prediction method based on historical viewpoint trajectory and the method based on K Nearest Neighbor (KNN) viewpoint prediction in work [26] (KNN-based viewpoint prediction, KVP) are selected for comparison. For a fair comparison, the K value in the cross-user viewpoint prediction method proposed in this paper is consistent with the K value in the work [26], that is, $K = 5$. Fig. 8 shows the trend of the accuracy of different viewpoint prediction methods along with the prediction time. It can be seen that the overall trend of the prediction accuracy of three methods is gradually decreases along with the time. The linear regression method performs the viewpoint prediction only based on the historical trajectory of the viewpoint. However, since the movement of the user's viewpoint changes according to the video content, the prediction accuracy based only on the historical trajectory of the viewpoint is the lowest, especially when the prediction time is long, the accuracy decreases faster. The KVP method can maintain a high accuracy rate when the prediction time is short. Similar to the linear regression method, its prediction accuracy rate decreases rapidly with time. Compared with the KVP method, the proposed method improves the accuracy by at least 6% on average, and the prediction accuracy can be maintained above 80% when the prediction time is long (i.e., 5 s).

In the next step, the proposed cache strategy is evaluated with

Table 1
Cache hit ratio with different video source.

Caching Strategy	Video1(%)	Video2(%)	Video3(%)
Proposed caching	**56**	**53**	**54**
Popularity based caching	33	31	32
Viewpoint prediction based caching	40	43	38

different video sources. Note that in the evaluation, we only calculate the cache hit rate of tile chunks for user requests in the evaluation, i.e., if a cached tile chunk is requested, it is recorded as a cache hit. We calculate the cache hit rate for different video sources (video1, video2, and video3) with different caching strategies. As shown in Table 1, we can see that our proposed caching strategy has the best cache hit rate for all three video sources, and the highest cache hit rate can reach 56%, which is remarkably helpful in improving the user experience.

Then, the proposed buffer management algorithm is evaluated with different bandwidth situations. Fig. 9 shows the fluctuations of user bandwidth records over time. It can be seen from the figure that the bandwidth fluctuation of trace 1 is small, and the bandwidth of trace 5 fluctuates greatly. With two kinds of bandwidth situation, HB strategy, TB strategy and TTB strategy are adopted for performance evaluation, respectively. And the utility value of the user in each time period is recorded. Fig. 10 shows the normalized average utility value of the users using the adaptive transmission mechanism of different strategies under two bandwidth fluctuations. It can be seen from Fig. 10 that under the two different bandwidth fluctuations, the average utility value of the HB strategy adaptive transmission is higher than the average utility value of the TB and TTB strategies. For small bandwidth variations (Trace1), the average utility of using HB strategy is approximately 18% and 12% higher than TB strategy and TTB strategy, respectively. For large bandwidth variations (Trace5), the average benefit of using HB strategy is approximately 20% and 13% higher than that of TB strategy and TTB strategy, respectively.

Fig. 11 shows the standard deviation of the normalized user utility value based on different strategies. From the figure, it can be seen that in general, the standard deviation of the user utility value of HB strategy is larger than the standard deviation of the user utility value of TB strategy and TTB strategy. In the case of small bandwidth fluctuations, the standard deviation of the user utility value of HB strategy is about 34% and 14% lower than the standard deviation of the user utility value of TB and TTB strategy, respectively. In the case of large bandwidth fluctuations, the standard deviation of the user utility value of HB strategy is approximately 48% and 25% lower than the standard deviation of the user utility of TB strategy and TTB strategy, respectively.

Fig. 12 shows the changes of normalized user utility values with different strategies over time. Among them, the left sub-figure (a) is the user utility value when the bandwidth fluctuation is small, and the right sub-figure (b) is the bandwidth fluctuation in the larger case. The user utility value, as can be seen from the figure, in the two kinds of bandwidth fluctuations, HB strategy can maintain a higher utility value, and its fluctuation is small. Based on the above analysis, it can be concluded that compared with the traditional buffer strategy TB and TTB, HB can improve the user's utility value, and maintain a relatively stable utility value fluctuation in the case of bandwidth fluctuations, that is, improve the user's QoE.

Furthermore, with different caching strategies and the proposed buffer update algorithm, we evaluate the user latency, as shown in Fig. 13, which shows the Cumulative Distribution Function (CDF) of latency. From this figure, it can be seen that with our proposed caching strategy, 90% of the user latency is within 20 ms, which basically meets

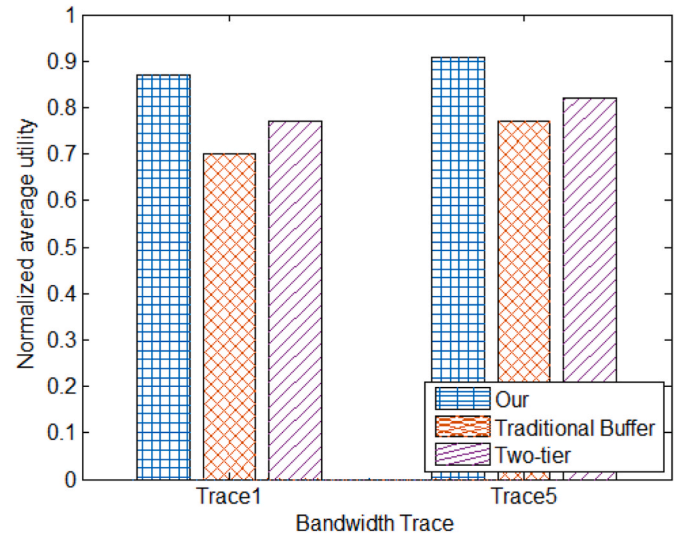


Fig. 10. Normalized average user utility value based on different strategies.

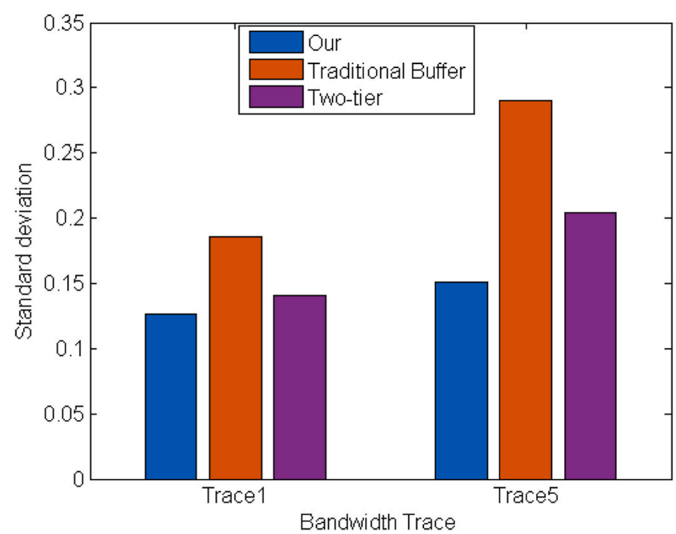
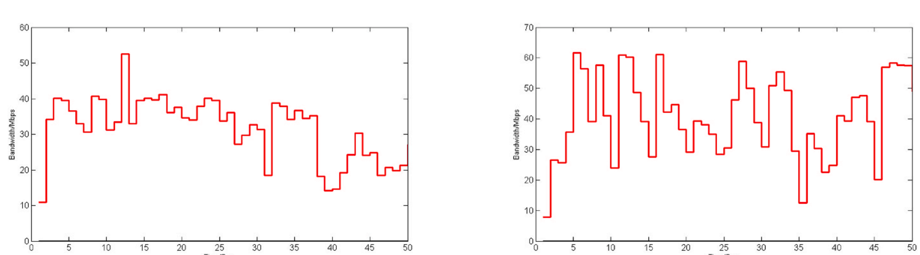


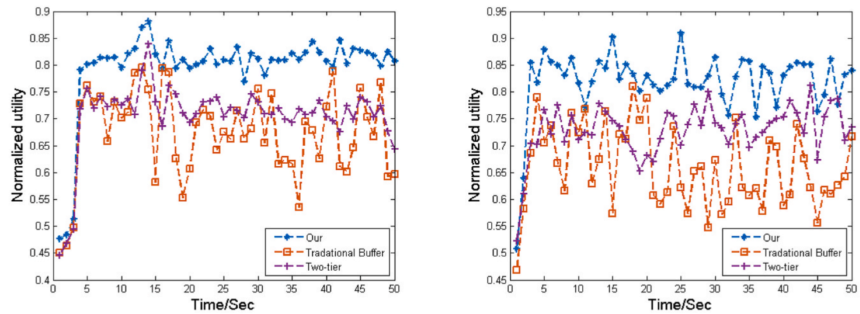
Fig. 11. Standard deviation of the normalized user utility.



(a) User bandwidth trace 1

(b) User bandwidth trace 5

Fig. 9. User bandwidth trace 1 and trace 5.



(a) User normalized utility value with small bandwidth fluctuations (b) User normalized utility value with large bandwidth fluctuations

Fig. 12. User normalized utility value based on different strategies.

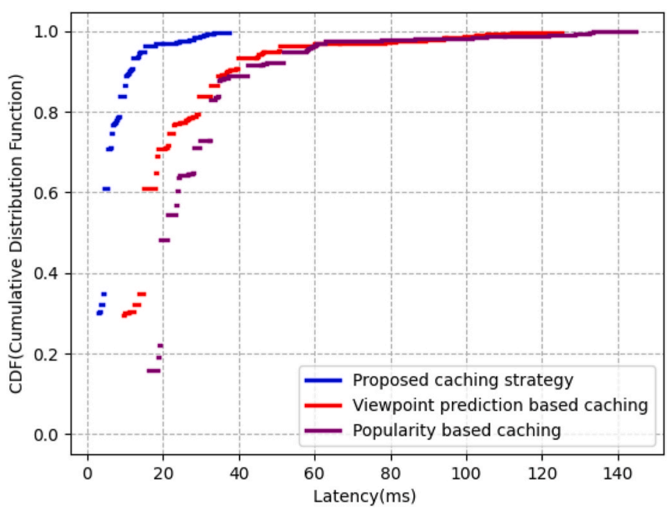


Fig. 13. The CDF of latency with different caching strategies.

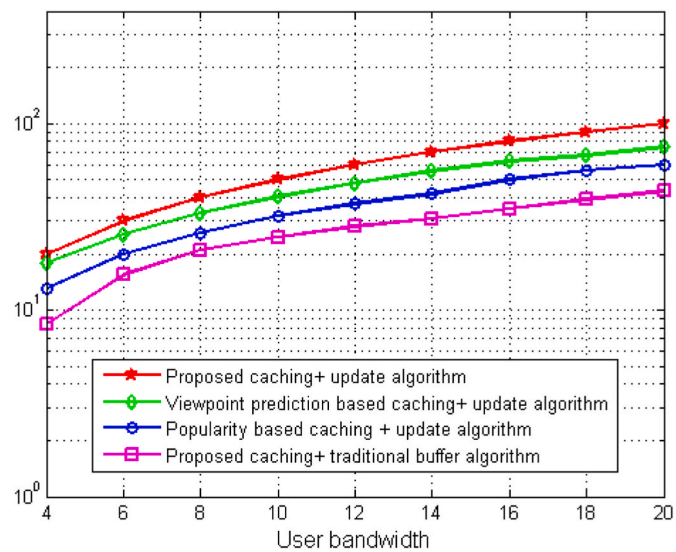


Fig. 14. QoE VS. User bandwidth with caching strategies and buffer update algorithms.

the VR latency requirements. In contrast, although the cache hit rate of the viewpoint prediction-based caching strategy is only about 10% lower than the hit rate of our proposed caching strategy, the total end-user latency exceeds 20 ms due to the large viewpoint prediction latency. In addition, we note that the popularity-based prediction, although without the predicted latency, however, has the worst user latency due to its lower cache hit ratio, where most content requests require a remote server or other MEC to respond.

Finally, we evaluate the caching strategy with the buffer update algorithm. In the simulation, we selected four different combinations of caching strategy + buffer update algorithms with different bandwidth condition, i.e., proposed caching + update algorithm, viewpoint prediction based caching + update algorithm, popularity based caching + update algorithm, proposed caching + traditional buffer algorithm. From Fig. 14, we can see that the user's QoE increases with the growth of the user bandwidth. Among them, the caching strategy + buffer update algorithm proposed in this paper achieves the highest user QoE. It is worth noting that the traditional buffer with the proposed caching strategy achieves the lowest QoE, which also verifies the importance of buffer management in another aspect.

6. Conclusion

The low latency and intensive computing requirements of mobile VR have brought new challenges to the development of 5G and its future mobile networks. To meet the demanding latency and bandwidth requirements of VR, the joint optimization of caching and buffer management is presented in this paper. First, a viewpoint prediction method based on historical trajectories and motion similarity between users is introduced, which significantly improves the viewpoint prediction accuracy over time. Based on 5G intelligent edge computing, we propose a tile popularity based on edge caching strategy, which VR content is proactively cached with tile granularity. Furthermore, an adaptive update algorithm based on hierarchical buffer is presented, which jointly considers the status of bandwidth, buffer and predicted viewpoint to update the tile chunk in the client's buffer. And the buffer update problem is modeled as an optimization problem, the corresponding solution algorithms are presented according to the decomposition of the problem. Finally, the simulation results show that the proposed proactive caching strategy cooperates with the hierarchical buffer update algorithm could achieve better QoE of mobile VR users.

Acknowledgements

This work is supported in part by the Chongqing Municipal Education Commission projects under Grant No. KJCX2020035, KJQN202200829, Chongqing Science and Technology Commission projects under grant No. CSTB2022BSXM-JCX0117 and cstc2020jcyj-msxmX0339, and is also supported in part by National Natural Science Foundation of China under Grant No. (62171072, 62172064, 62003067, 61901067), and is also supported in part by Chongqing Technology and Business University projects under Grant no. (2156004, 212017).

References

[1] S. Xia, Z. Yao, Y. Li, S. Mao, Online distributed offloading and computing resource management with energy harvesting for heterogeneous mec-enabled iot, IEEE Trans. Wireless Commun. 20 (10) (2021) 6743–6757.
[2] J. Yang, F. Lin, C. Chakraborty, K. Yu, Z. Guo, A. Nguyen, J.J.P.C. Rodrigues, A parallel intelligence-driven resource scheduling scheme for digital twins-based intelligent vehicular systems, IEEE Trans. Intell. Veh. 8 (4) (2023) 2770–2785.
[3] Z. Zhou, X. Dong, Z. Li, K. Yu, C. Ding, Y. Yang, Spatio-temporal feature encoding for traffic accident detection in VANET environment, IEEE Trans. Intell. Transport. Syst. 23 (10) (2022) 19772–19781.
[4] K. Yu, L. Tan, C. Yang, K.R. Choo, A.K. Bashir, J.J.P.C. Rodrigues, T. Sato, A blockchain-based shamir's threshold cryptography scheme for data protection in industrial internet of things settings, IEEE Internet Things J. 9 (11) (2022) 8154–8167.
[5] Z. Guo, K. Yu, Z. Lv, K.R. Choo, P. Shi, J.J.P.C. Rodrigues, Deep federated learning enhanced secure POI microservices for cyber-physical systems, IEEE Wireless Commun. 29 (2) (2022) 22–29.
[6] Z. Guo, K. Yu, N. Kumar, W. Wei, S. Mumtaz, M. Guizani, Deep-distributed-learning-based POI recommendation under mobile-edge networks, IEEE Internet Things J. 10 (1) (2023) 303–317.
[7] B. Zhu, K. Chi, J. Liu, K. Yu, S. Mumtaz, Efficient offloading for minimizing task computation delay of noma-based multiaccess edge computing, IEEE Trans. Commun. 70 (5) (2022) 3186–3203.
[8] C. Chen, Z. Liao, Y. Ju, C. He, K. Yu, S. Wan, Hierarchical domain-based multi-controller deployment strategy in sdn-enabled space-air-ground integrated network, IEEE Trans. Aero. Electron. Syst. 58 (6) (2022) 4864–4879.
[9] M. Chen, K. Hu, I. Chung, C. Chou, Towards VR/AR multimedia content multicast over wireless LAN, in: Proceedings of the 16th IEEE Annual Consumer Communications & Networking Conference, CCNC, 2019, IEEE, 2019, pp. 1–6.
[10] F. Duanmu, E. Kurdoglu, S.A. Hosseini, Y. Liu, Y. Wang, Prioritized buffer control in two-tier 360 video streaming, in: Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, VR/AR Network@SIGCOMM 2017, ACM, 2017, pp. 13–18.
[11] M. Chen, W. Saad, C. Yin, Virtual reality over wireless networks: quality-of-service model and learning-based resource management, IEEE Trans. Commun. 66 (11) (2018) 5621–5635.
[12] M. Almquist, V. Almquist, V. Krishnamoorthi, N. Carlsson, D.L. Eager, The prefetch aggressiveness tradeoff in 360° video streaming, in: Proceedings of the 9th ACM Multimedia Systems Conference, MMSys 2018, ACM, 2018, pp. 258–269.
[13] F. Duanmu, E. Kurdoglu, S.A. Hosseini, Y. Liu, Y. Wang, Prioritized buffer control in two-tier 360 video streaming, in: Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, VR/AR Network@SIGCOMM 2017, ACM, 2017, pp. 13–18.
[14] M. Budagavi, J. Furton, G. Jin, A. Saxena, J. Wilkinson, A. Dickerson, 360 degrees video coding using region adaptive smoothing, in: 2015 IEEE International Conference on Image Processing, ICIP 2015, IEEE, 2015, pp. 750–754.
[15] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masiá, G. Wetzstein, Saliency in VR: how do people explore virtual environments? IEEE Trans. Vis. Comput. Graph. 24 (4) (2018) 1633–1642.
[16] P. Rondao-Alface, J. Macq, N. Verzijp, Interactive omnidirectional video delivery: a bandwidth-effective approach, Bell Labs Tech. J. 16 (4) (2012) 135–147.
[17] A.T. Nasrabadi, A. Mahzari, J.D. Beshay, R. Prakash, Adaptive 360-degree video streaming using scalable video coding, in: Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, ACM, 2017, pp. 1689–1697.
[18] L. Feng, Z. Yang, Y. Yang, X. Que, K. Zhang, Smart mode selection using online reinforcement learning for VR broadband broadcasting in D2D assisted 5g hetnets, IEEE Trans. Broadcast. 66 (2) (2020) 600–611.
[19] J. Yang, J. Luo, D. Meng, J. Hwang. Qoe-driven resource allocation optimized for delay-sensitive VR video uploading over cellular network, IEEE, Barcelona, Spain, 2019, pp. 1–6.
[20] Y. Li, H. Ma, L. Wang, S. Mao, G. Wang, Optimized content caching and user association for edge computing in densely deployed heterogeneous networks, IEEE Trans. Mobile Comput. 21 (6) (2022) 2130–2142.
[21] M. Abdelrahman, M. Elbamby, V. Räisänen, Proactive scheduling and caching for wireless VR viewport streaming, in: IEEE Globecom 2021 Workshops, IEEE, 2021, pp. 1–6.
[22] J. Yang, J. Luo, J. Wang, S. Guo, Cmu-Vp, Cooperative multicast and unicast with viewport prediction for VR video streaming in 5g H-CRAN, IEEE Access 7 (2019) 134187–134197.
[23] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, Y. Wang, CUB360: exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming, in: Proceedings of the 2018 IEEE International Conference on Multimedia and Expo, ICME 2018, IEEE Computer Society, 2018, pp. 1–6.
[24] A.D. Abreu, C. Ozcinar, A. Smolic, Look around you: saliency maps for omnidirectional images in VR applications, in: Proceedings of the Ninth International Conference on Quality of Multimedia Experience, QoMEX 2017, IEEE, 2017, pp. 1–6.
[25] G. Ravindra, W.T. Ooi, On tile assignment for region-of-interest video streaming in a wireless LAN, in: Proceedings of the 2012 Network and Operating System Support for Digital Audio and Video Workshop, ACM, 2012, pp. 59–64.
[26] S. Liao, J. Wu, J. Li, K. Konstantin, Information-centric massive iot-based ubiquitous connected VR/AR in 6g: a proposed caching consensus approach, IEEE Internet Things J. 8 (7) (2021) 5172–5184.
[27] C. Wu, Z. Tan, Z. Wang, S. Yang, A dataset for exploring user behaviors in VR spherical video streaming, in: Proceedings of the 8th ACM on Multimedia Systems Conference, ACM, 2017, pp. 193–198.
[28] Q. Cheng, H. Shan, W. Zhuang, L. Yu, Z. Zhang, T.Q.S. Quek, Design and analysis of MEC- and proactive caching-based 360° mobile VR video streaming, IEEE Trans. Multimed. 24 (2022) 1529–1544.
[29] J. Park, J. Hwang, H. Wei, Cross-layer optimization for VR video multicast systems, in: Proceedings of the IEEE Global Communications Conference, IEEE, 2018, pp. 206–212.
[30] Y. Jin, J. Liu, F. Wang, Ebublio: edge assisted multi-user 360-degree video streaming, in: Proceedings of the 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, VR Workshops, IEEE, 2022, pp. 600–601.
[31] K. Liu, Y. Liu, J. Liu, A. Argyriou, Tile caching for scalable VR video streaming over 5g mobile networks, J. Vis. Commun. Image Represent. 79 (2021) 103210.
[32] J. Dai, Z. Zhang, S. Mao, D. Liu, A view synthesis-based 360° VR caching system over mec-enabled C-RAN, IEEE Trans. Circ. Syst. Video Technol. 30 (10) (2020) 3843–3855.
[33] M. Peng, T.Q.S. Quek, G. Mao, Z. Ding, C. Wang, Artificial-intelligence-driven fog radio access networks: recent advances and future trends, IEEE Wireless Commun. 27 (2) (2020) 12–13.
[34] P. Maniotis, E. Bourtsoulatze, N. Thomos, Tile-based joint caching and delivery of 360° videos in heterogeneous networks, IEEE Trans. Multimed. 22 (9) (2020) 2382–2395.
[35] Y. Liu, J. Liu, A. Argyriou, L. Wang, Z. Xu, Rendering-aware VR video caching over multi-cell MEC networks, IEEE Trans. Veh. Technol. 70 (3) (2021) 2728–2742.
[36] F. Guo, F.R. Yu, H. Zhang, H. Ji, V.C.M. Leung, X. Li, An adaptive wireless virtual reality framework in future wireless networks: a distributed learning approach, IEEE Trans. Veh. Technol. 69 (8) (2020) 8514–8528.
[37] S. Zehtabian, M. Razghandi, L. Bölöni, D. Turgut, Predictive caching for AR/VR experiences in a household scenario, in: Proceedings of the 2020 International Conference on Computing, Networking and Communications, IEEE, 2020, pp. 591–595.
[38] H. Xiao, C. Xu, Z. Feng, R. Ding, S. Yang, L. Zhong, J. Liang, G. Muntean, A transcoding-enabled 360° VR video caching and delivery framework for edge-enhanced next-generation wireless networks, IEEE J. Sel. Area. Commun. 40 (5) (2022) 1615–1631.
[39] H. Ahmadi, O. Eltobgy, M. Hefeeda, Adaptive multicast streaming of virtual reality content to mobile users, in: Proceedings of the on Thematic Workshops of ACM Multimedia 2017, ACM, 2017, pp. 170–178.
[40] G. Papaioannou, I. Koutsopoulos, Tile-based caching optimization for 360° videos, in: Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM, 2019, pp. 171–180.