Joint EPC and RAN Caching of Tiled VR Videos for Mobile Networks

Kedong Liu[1,2], Yanwei Liu[1,2] *, Jinxia Liu[3], Antonios Argyriou[4], and Ying Ding[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Zhejiang Wanli University, Ningbo, China
[4] University of Thessaly, Volos, Greece

Abstract. In recent years, 360-degree VR (Virtual Reality) video has brought an immersive way to consume content. People can watch matches, play games and view movies by wearing VR headsets. To provide such online VR video services anywhere and anytime, the VR videos need to be delivered over wireless networks. However, due to the huge data volume and the frequent viewport-updating of VR video, its delivery over mobile networks is extremely difficult. One of the difficulties for the VR video streaming is the latency issue, i.e., the necessary viewport data cannot be timely updated to keep pace with the rapid viewport motion during viewing VR videos. To deal with this problem, this paper presents a joint EPC (Evolved Packet Core) and RAN (Radio Access Network) tile-caching scheme that pushes the duplicates of VR video tiles near the user end. Based on the predicted viewport-popularity of the VR video, the collaborative tile data caching between the EPC and RAN is formulated as a 0-1 knapsack problem, and then solved by a genetic algorithm (GA). Experimental results show that the proposed scheme can achieve great improvements in terms of the saved transmission bandwidth as well as the latency over the scheme of traditional full-size video caching and the scheme that the tiles are only cached in the EPC.

Keywords: VR · Cache · EPC · RAN · Video tiles.

1 Introduction

Recently, the 360-degree VR video applications are becoming more and more popular with the increasing maturity of VR technology. At the same time, the rapid development of wireless communication has made it possible to distribute 360-degree VR videos over wireless networks.

To create an immersive experience for the end users, panoramic VR video provides a 360×180 degree field of view with a high resolution (4K or beyond), and thus usually tends to consume a large amount of storage space and transmission bandwidth. Furthermore, due to the particularly interactive nature of the

* Corresponding author. This work was supported in part by National Natural Science Foundation of China under Grants 61771469 and 61572497, and Zhejiang Provincial Natural Science Foundation of China under Grant LY17F010001.

viewport data delivery, VR video systems have very strict latency requirements [11]. This brings a great pressure on the network especially the wireless part. It is quite challenging to transmit VR videos over mobile networks.

Since VR video consumes bandwidth, a number of VR video coding and transmission approaches were proposed by researchers to reduce the data volume by applying source data compression. In [4], a region-adaptive video smoothing approach was proposed to improve the encoding efficiency by considering the particular characteristics of sphere-to-plane projection. To enhance the ability of spatial random access, VR video tiling was also used during streaming. In [7], Gaddam *et al.* applied a tiling scheme to deliver different quality levels for different parts of the panoramic VR videos. In [14], Skupin *et al.* proposed an alternative approach to 360° video facilitating HEVC (High Efficiency Video Coding) tiles. To reduce the necessary data amount for the user, an approach was presented by Guntur *et al.* in [8] to transmit the tiled regions of a video to support RoI (Region of Interest) streaming. By taking a step further, Corbillon *et al.* in [5] proposed a viewport-adaptive 360-degree video streaming system to transmit VR videos by reducing the transmitted bit-rates of tiles. From the video networking perspective, the Dynamic Adaptive Streaming over HTTP (DASH) for 360-degree VR videos can also reduce the transmitted VR video data [9, 10]. The above-mentioned approaches can reduce the transmitted VR video data amount significantly. However, due to multi-user concurrent requests, current VR video applications still consume higher bandwidth that incurs large transmission delay.

To deal with the latency issue of video streaming, video caching has been proposed to push duplicate videos near the user ends. This way can reduce the duplicate content transmissions and relieve the pressure on mobile networks as well. In [16], Xie *et al.* studied the effects of different access types on Internet video services and their implications on Content Delivery Network (CDN) caching. Franky *et al.* in [6] studied a video cache system which can reduce the video traffic and the loading time. In [18], Zhou *et al.* proposed a QoE-driven video cache allocation scheme for mobile cloud server. These methods are very effective in reducing the delivery latency in the fixed broadband networks, but in mobile networks, they cannot achieve the same results.

To further reduce the latency, cache servers can be deployed to the RAN that is closest to the user end. In [15], Wang *et al.* studied the caching techniques for both the EPC and RAN. In [12], Shen *et al.* designed an information-aware QoE-centric mobile video cache scheme. In [1], Ahlehagh *et al.* introduced a video-aware caching scheme in the RAN. Ye *et al.* in [17] studied the quality-aware DASH video caching scheme at mobile network edge. These approaches can further reduce the video streaming latency by caching the content in mobile networks. However, they neglected the collaboration between the EPC and RAN during the video data caching. In addition, these video caching approaches were originally designed for full-size videos and they cannot efficiently work for the VR videos due to the particular characteristic of VR videos, i.e., the tremendous size of video data, which might take up too much cache space.

Usually, people watch only a part of the VR video spatially not the full-size video, and thus the VR video can be cached with the tiled-chunk representation to reduce the occupied cache space. The VR video sequence is first segmented into several tiles spatially and then a number of chunks temporally. The tiled-chunk data is deployed in the EPC and RAN beforehand according to the prediction of user's viewport popularity. Additionally, taking account of the differences in transmission distance between the RAN cache and the EPC cache, the joint RAN and EPC caching scheme needs to designed. On the one hand, caches in the RAN are close to the end-user which can save the content transmission latency and relieve the bandwidth pressure for the backhaul network. However, the cache space in the RAN is strictly limited and each eNodeB (evolved NodeB) in the RAN may only serve a few users, which results in low cache hit rate in some cases. On the other hand, caches in the EPC aggregate many UEs (User Equipments) and the cache hit rate is higher, but it will incur higher latency than that in the RAN. To deal with these issues mentioned above, we propose to tile the VR video and cache the tiled VR video chunks in the EPC and RAN collaboratively.

By making full consideration of the characteristics of VR videos and the architecture of mobile networks, this paper presents a joint EPC and RAN tile-caching scheme for mobile networks. The contributions of this paper are summarized below.

- Taking into account the fact that only a small portion of the complete 360-degree VR video is visible to a viewer, a tile-caching scheme is proposed. By segmenting VR videos into several tiles spatially and a series of chunks temporally, the tiles within the users' viewports are more likely to be cached than the tiles out of the viewports. This can significantly save the cache space compared to the full-size video caching strategy.
- To reduce the user-perceived latency as well as the redundant traffic over the network, caches are deployed in both the EPC and RAN. Moreover, the joint EPC and RAN tile-caching scheme is proposed to maximize the saved system bandwidth cost subject to the constraint of viewport-requesting latency. The caching optimization process is formulated as a 0-1 knapsack problem, and then solved by a GA.

The rest of the paper is organized as follows. In Section 2, the proposed joint EPC and RAN tile-caching scheme is described. Experimental results are provided in Section 3. Finally, Section 4 concludes the paper.

2 Joint EPC and RAN Tile-Caching Scheme

The proposed joint EPC and RAN tile-caching scheme is shown in Fig. 1. Based on the architecture of mobile networks, the EPC and each RAN are equipped with a cache respectively, and they are regarded as the cache nodes. The cache in the EPC is deployed in the Packet Data Network Gateway (P-GW) and the caches for each RAN are deployed in eNodeBs. In addition, there is a logically centrally-deployed entity (Content Controller) which is connected to the P-GW. The Content Controller is responsible for recognizing VR video request from the UEs and then performs the caching optimization algorithm in terms of the

collected information from each cache node. To improve the cache hit rate for different tiles in the video, a collaborative caching approach is used to optimize the caching placement of tiles among the EPC and RAN. The cache nodes cache the VR video tiles based on the optimization computation results.

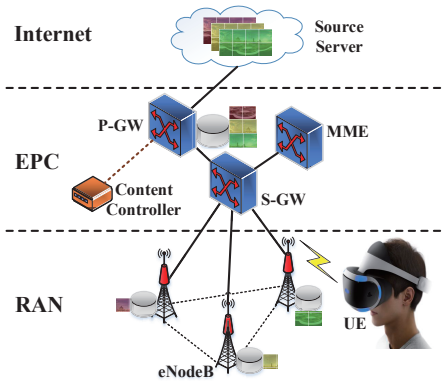


Fig. 1. Joint EPC and RAN tile-caching scheme.

In the optimization, the VR video tiles within users' viewports are more likely to be cached near the UE. Once a viewer requests a VR video viewport using a UE, the eNodeB will check whether the requested viewports were already existed in the RAN cache. If the requested data is available, the RAN cache node will serve the request and the requested VR video tiles will be transmitted to the UE through the wireless radio access network. If the requested data is not available locally in the RAN, the request will be transferred to the Content Controller to check whether the EPC and the other RAN cache nodes have had already cached the requested VR video tiles. If cached, the VR video tiles will be transmitted to the corresponding eNodeB through wired connections from EPC cache node or through wireless connections (e.g., interface X2 [15]) from the other RAN cache nodes, and finally transmitted to the UE. If none of the cache nodes had cached the requested VR video viewports, the request can only be served by the source server on the Internet.

2.1 Tile-caching problem formulation

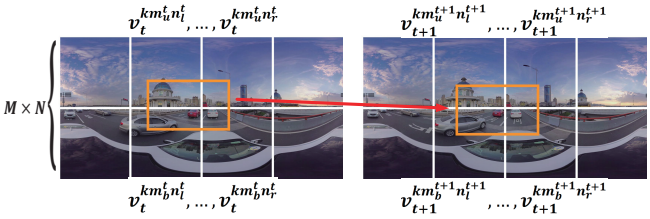


Fig. 2. Tile partition and viewport moving.

According to the limitation of the field of view (usually 120 degrees) for human eyes, only a small part of the full frame VR image is watched in one moment which is called the viewport. That means only the VR video tiles within the viewport can be displayed on UE for watching. As shown in Fig. 2, the areas in the orange rectangles are frequently watched by the users that they can be

predicted in terms of the popularity of viewport. The popularity of viewport in the whole image is obtained via the saliency map prediction approach [13].

We define $\mathcal{T} = \{0, 1, ..., t, ..., T\}$ the set of the time slots and $\mathcal{K} = \{1, 2, ..., k, ..., K\}$ the set of VR videos. For a VR video k, $M \times N$ VR video tiles were obtained after the tiling process. v_t^{kmn} denotes the VR video tile at the mth row and nth column ($0 < m \leq M$, $0 < n \leq N$) in the kth video at the time slot t. Similarly, in the temporal dimension, the tiles were also divided into many chunks. Because the user's viewport varies with time and one chunk is with very short time, we can use an enlarged and unchanged viewport to denote the viewports for all frames in the whole chunk. The request from UE for the VR video k at the time of t denotes the request for a set of VR video tiles V_t^{kmn} covered by the enlarged viewport ($m_u^t \leq m \leq m_b^t$, $n_l^t \leq n \leq n_r^t$) at the time of t. m_u^t, m_b^t, n_l^t and n_r^t denote the tile number of the up row, bottom row, left column and right column that the viewport occupies, respectively. As a consequence, once the set of VR video tiles V_t^{kmn} are cached, the whole VR video is supposed to be cached technically because the tiles out of the viewport in the chunk are usually not necessarily watched.

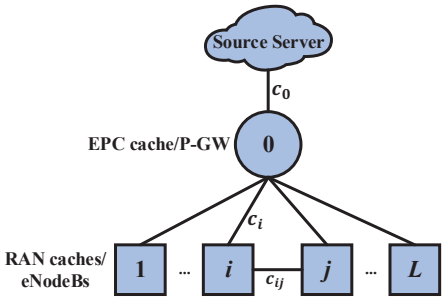


Fig. 3. The joint EPC and RAN tile-caching network architecture.

In the joint EPC and RAN tile-caching scheme, caches are deployed inside both the EPC (P-GW) and the RAN (eNodeB). The caching network architecture is abstracted as the graph in Fig. 3. Denote c_i as the unit cost for transferring VR video tiles from the P-GW to eNodeB i, c_0 as the unit cost when transferring VR video tiles from the source server to P-GW and c_{ij} as the unit cost when transferring VR video tiles between eNodeBs i and j. To formulate the tile-caching problem, the transmission bandwidth cost of VR videos is utilized as the optimization metric. The optimization goal of the scheme is to minimize the total bandwidth cost for serving all VR video requests subjecting to the overall disk storage limitation of cache nodes and the system latency constraint. Easily, the problem can be transformed into an equivalent problem of maximizing the saving cost subjecting to the cache space limitation and latency constraint compared to the way that obtains VR video tiles from the source server.

Denote the 0-1 variable $x_{t,i}^{kmn}$ as the indication of whether the VR video tile v_t^{kmn} is cached in the cache node i. If node i had already cached the VR video tile v_t^{kmn}, $x_{t,i}^{kmn} = 1$; otherwise $x_{t,i}^{kmn} = 0$. Based on the above definition, there are basically four ways to fetch a VR video for viewers:

- If the cache node eNodeB i can fulfill the request from the UE locally for the VR video tile v_t^{kmn}, the unit cost saving is $c_0 + c_i$.
- If the request cannot be fulfilled locally by eNodeB i but can be fulfilled by the other eNodeBs, e.g., the node j ($i \neq j$), the unit cost saving can be written as $c_0 + c_i - c_{ij}$.
- If the request can be fulfilled by the EPC cache at the P-GW, the unit cost saving is c_0.
- If the request can only be fulfilled from the source server on the Internet, the unit cost saving is 0.

In the following, we define the saved bandwidth cost $P_{t,i}^{kmn}$ when the request for the VR video tile v_t^{kmn} at node i is fulfilled by the EPC cache. $P_{t,i}^{kmn}$ is given by

$$P_{t,i}^{kmn} = c_0 \times x_{t,0}^{kmn}. \tag{1}$$

Also, the maximal saved cost $Q_{t,ij}^{kmn}$ when the request for VR video tile v_t^{kmn} at node i is fulfilled by another eNodeB j is defined as

$$Q_{t,ij}^{kmn} = \max_{j \in \mathcal{L} \setminus \{i\}} \{(c_0 + c_i - c_{ij}) y_{t,ij}^{kmn}\}, \tag{2}$$

where $\mathcal{L}$ is the set of the cache nodes which can be expressed as $\mathcal{L} = \{0, 1, .., i, ..., j, ..., L\}$, and $y_{t,ij}^{kmn}$ is also a 0-1 variable which indicates whether the request for the VR video tile v_t^{kmn} from the UE connecting to eNodeB i is transferred to eNodeB j.

Based on the above analysis, the total saved cost τ for UEs compared to the way that obtains VR video from the source server can be calculated as:

$$\begin{aligned}\tau &= \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \tau_k(\boldsymbol{X_t}) \\ &= \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{L}} \sum_{m_u^t \leq m \leq m_b^t} \sum_{n_l^t \leq n \leq n_r^t} \lambda_{t,i}^{kmn} \cdot s_t^{kmn} \cdot [x_{t,i}^{kmn} \cdot \\ &\quad (c_0 + c_i) + (1 - x_{t,i}^{kmn}) \cdot \max\{P_{t,i}^{kmn}, Q_{t,ij}^{kmn}\}],\end{aligned} \tag{3}$$

where $\tau_k(\cdot)$ is a function to calculate the saved bandwidth cost for the VR video k. $\boldsymbol{X_t}$ is a set of 0-1 variable $x_{t,i}^{kmn}$ that denotes the caching result of a VR video k at the time of t, and $\boldsymbol{X_t}$ can be expressed as

$$\boldsymbol{X_t} = (x_{t,0}^{k11}, x_{t,0}^{k12}, ..., x_{t,0}^{kmn}, ..., x_{t,i}^{kmn}, ..., x_{t,L}^{kMN}). \tag{4}$$

where s_t^{kmn} denotes the file size of the VR video tile v_t^{kmn}. The request probability $\lambda_{t,i}^{kmn}$ for the VR video tile v_t^{kmn} from the UE connecting to eNodeB i is given by

$$\lambda_{t,i}^{kmn} = \xi_i^k \cdot \theta_t^{kmn}, \tag{5}$$

where ξ_i^k indicates the probability of requesting for the VR video k from the UE connecting to eNodeB i. θ_t^{kmn} denotes the probability of requesting for the tile v_t^{kmn} in VR video k, which can be obtained from the viewport popularity data of the VR videos.

Finally, the tile caching optimization problem of maximizing the saving cost τ can be mathematically formulated as

$$\max_{\boldsymbol{X_t}} \tau \tag{6}$$

$$s.t. \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{m \in \{1,2,...,M\}} \sum_{n \in \{1,2,...,N\}} s_t^{kmn} x_{t,i}^{kmn} \leq B_i \tag{7}$$

$$\begin{aligned} x_{t,i}^{kmn} &\in \{0, 1\}, \forall i \in \mathcal{L}, t \in \mathcal{T}, k \in \mathcal{K}, \\ &m \in \{1, 2, ..., M\}, n \in \{1, 2, ..., N\} \end{aligned} \tag{8}$$

$$\begin{cases} \max\limits_{i} \left\{ \frac{x_{t,i}^{kmn} \cdot s_t^{kmn}}{w_i} \right\} \leq T, & \text{when} \sum\limits_{i} x_{t,i}^{kmn} \neq 0, \\ \frac{s_t^{kmn}}{w_s} \leq T, & \text{otherwise}, \\ \forall t \in \mathcal{T}, k \in \mathcal{K}, m_u^t \leq m \leq m_b^t, n_l^t \leq n \leq n_r^t, \end{cases} \tag{9}$$

where B_i denotes the cache space of the cache node i, w_i and w_s denote the available bandwidth from RAN cache node i to the UE and from the source server to the UE, respectively. T denotes the maximum limitation of transmission latency.

We know that, constraint (7) is used for cache space optimization. It guarantees that the space which the cached VR video tiles occupies doesn't exceed the cache space limitation. Constraint (8) indicates that the VR video tiles cannot be further divided anymore. 1 and 0 denote whether the cache node cached the VR video tile or not, respectively. Constraint (9) shows that the request for the tiles within the user's viewport should be responded and fulfilled timely under the constraint of transmission latency T. Specifically, the latency for transmitting the requested VR video tiles should be less than or equal to the maximum limitation of transmission latency T. In the caching system, the delivery distances for the tiles that the viewport covers are different because they are probably located in different cache nodes. Obviously, the delivery latency for the viewport depends on the maximum delivery latency for all the tiles within the user's viewport.

2.2 Solution

Algorithm 1 GA for the joint EPC and RAN tile-caching optimization

Input: The population size s_{pop}, the chromosome length l, the probability of performing crossover p_c, probability of mutation p_m and the termination number of generations n_{ge}.

Output: The optimal caching result $\boldsymbol{X}$.

1: Initialization: generate the population of $\boldsymbol{X}$. The number of generation $g \leftarrow 0$.
2: **repeat**
3: **Selection:** calculate the fitness function according to Eq. (3), specially $\tau \leftarrow 0$ if the $\boldsymbol{X}$ cannot satisfy the constraints.
4: **Sort** the individuals according to τ in a decending order and select a portion of population using roulette wheel selection to breed a new generation.
5: **Crossover:** update p_c, calculate the number of crossover $s_{pop} \times p_c$, and do the crossover operation to generate a new generation.
6: **Mutation:** calculate the number of mutation $s_{pop} \times p_m$, and mutate to generate a second generation.
7: $g \leftarrow g + 1$.
8: **until** $g = n_{ge}$.
9: **return** $\boldsymbol{X}$ of the highest τ.

Based on the formulations from (6) to (9), the tile-caching problem is in line with the definition of the 0-1 knapsack problem. Due to its combinatorial nature, 0-1 knapsack is a NP-hard problem. As we all know, the GA has the advantage of the global optimization and the parallelism in seeking the solutions to the optimization problem, which indicates the solution-searching process can be implemented in parallel. Thus, to find the final result of placing VR video

tiles in the cache nodes, we adopt the GA, a kind of heuristic algorithm, to solve the proposed optimization problem.

In the GA for joint EPC and RAN tile-caching optimization, the final optimization result $\boldsymbol{X}$ that has the highest fitness value is a set of $\boldsymbol{X_t}$ ($t \in \mathcal{T}$) for all the videos in $\mathcal{K}$. To represent the solution space in GA, we use the binary coding string $\boldsymbol{X}$ as the chromosome. The chromosome length l denotes the number of the 0-1 variables $x_{t,i}^{kmn}$ in one of the solution results $\boldsymbol{X}$. Firstly, the population size s_{pop}, the chromosome length l, the probability of performing crossover p_c, probability of mutation p_m and the termination criteria (the fixed number of generations n_{ge}) are initialized. Then, the first generation of population is initialized by generating the candidate solutions of the caching result $\boldsymbol{X}$. Next, the fitness value τ of each population $\boldsymbol{X}$ is calculated in terms of Eq. (3). If the individual $\boldsymbol{X}$ doesn't satisfy the constraints (7), (8) or (9), the fitness value τ will be zero. In step 4, roulette wheel selection is used to select a portion of population to breed a new generation. In order to avoid the problem of premature convergence, scale factor is introduced to update p_c in step 5 [2]. In steps 5-6, the operations of crossover and mutation are performed to generate a second generation. Finally, after n_{ge} loops, we can get the optimal caching result $\boldsymbol{X}$. Since the GA belongs to a non-deterministic class of algorithms, the optimal solution may vary for each run of the algorithm with the same input parameters. Thus the final result $\boldsymbol{X}$ is rather sub-optimal. The specific GA for joint EPC and RAN tile-caching optimization is shown in Algorithm 1.

3 Experimental Results

3.1 Experimental setup

Table 1. Experimental parameters

Tile size	**Viewport size**	**Chunk length**	**RAN cache number (L)**	**Cache size in eNodeB**	**UE number per eNodeB**	$\boldsymbol{T}$	$\boldsymbol{c_0}$	$\boldsymbol{c_i}$
960×960	1920×1080	$1s$	40	$10G$	100	$15ms$	100	5
$\boldsymbol{c_{ij}}$	$\boldsymbol{w_i}$	$\boldsymbol{w_s}$	$\boldsymbol{s_{pop}}$	$\boldsymbol{l}$	$\boldsymbol{p_c}$	$\boldsymbol{p_m}$	$\boldsymbol{n_{ge}}$	
2∼10	$600Mbps$	$150Mbps$	50	2000	0.7∼0.9	0.02	500	

To evaluate the proposed joint EPC and RAN tile-caching scheme, we developed a custom software in Java to realize the optimization algorithm. HEVC reference software HM 15.0 was used to encode the VR videos. The five 360-degree VR video test sequences with spatial size of 3840×1920 (AerialCity, DrivingInCity, DrivingInCountry, Harbor and PoleVault_le) were obtained from JVET [3]. They were divided into 4×2 tiles for the caching optimization scheme. The popularity of the VR videos (ξ in Eq. (5)) is assumed to follow a Zipf popularity distribution and the VR video k is requested with the probability $\xi^k = \beta/k^\alpha$, where $\beta = (\sum_{k=1}^{K} k^{-\alpha})^{-1}$. The Zipf parameter α was initialized as 0.75. The capacity ratio, which means the ratio of the aggregate size of video tiles to the total cache size was set to 60%. The key experimental parameters are shown in Table 1. To verify the performance of the proposed Joint EPC and RAN Tile-Caching (JERTC) scheme, we compared the proposed JERTC scheme with the scheme of Full-size VR video Caching without tiling (FC). Also, the Only EPC Caching (OEC) scheme was compared with the FC scheme. As the benchmark

scheme, the FC scheme is based on the well-known Least Recently Used (LRU) caching algorithm [1].

3.2 An illustration of the caching optimization result

Fig. 4 illustrates one example of the caching optimization result. $\boldsymbol{X_t}$ is an example extracted from the optimization result $\boldsymbol{X}$ for the VR video k at the time of t. 0 means that the corresponding video tile should not be cached in the cache node i. On the contrary, the video tile marked 1 should be cached in the cache node i. It can be seen from Fig. 4 that the tiles within the viewport are more probable to be cached locally in the wireless access network.

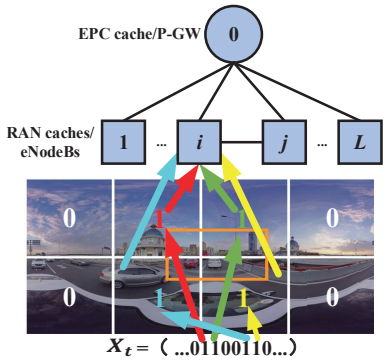
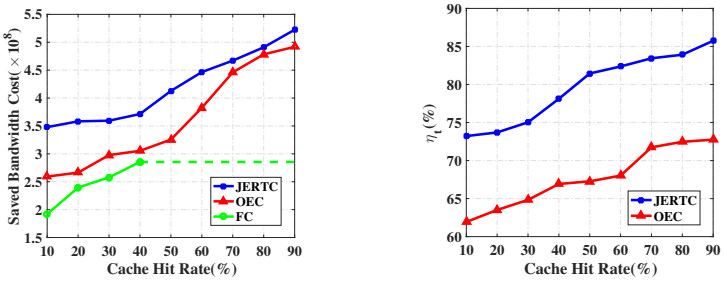


Fig. 4. One example of tile placement in the ith eNodeB.

3.3 Bandwidth and latency performances



(a) (b)

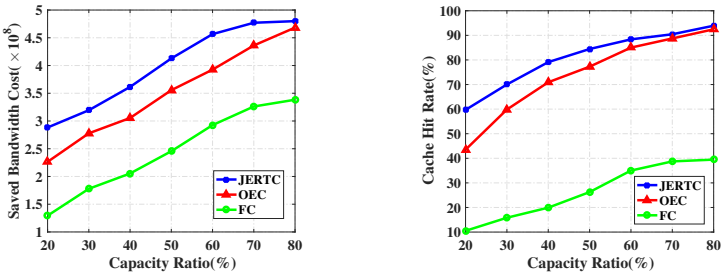
Fig. 5. (a) The curves of the saved bandwidth cost vs. the cache hit rate for the JERTC, OEC and FC scheme. (b) The curves of the saved latency vs. the cache hit rate for the JERTC scheme and the OEC scheme against the FC scheme.

Fig. 5(a) shows the saved bandwidth cost curves with the increasing cache hit rates for the JERTC scheme, the OEC scheme and the FC scheme. Due to the limitation of the cache space in the eNodeB, the cache hit rate of the FC scheme can reach only to about 40% ($\alpha = 0.75$, capacity ratio is 60%). It can be seen from Fig. 5(a) that the proposed JERTC scheme can save more bandwidth cost than the OEC scheme at the same cache hit rate. It highlights the great effectiveness and advantages of the JERTC scheme against the OEC and FC. With the increasing of the cache hit rate, all the three schemes can save more bandwidth because more VR video tiles were found in the cache nodes in the

mobile network. Besides, in Fig. 5(a) the gap between JERTC and OEC curves at the low cache hit rate is larger than that at high cache hit rate. With the increasing cache hit rate, the gap between the two schemes is gradually reduced. This is because at low cache hit rate, the requests from UE are mostly served by the source server besides the EPC cache node for the OEC scheme, and comparably most of the requests are served by RAN cache nodes and the EPC cache node for the proposed scheme. Consequently, the OEC scheme consumes more bandwidth than the proposed scheme at low cache hit rate. In contrast, at high cache hit rate, only a small part of the requests need to be served by the source server for the OEC scheme. Thus, a narrowing gap between the two schemes arises at high cache hit rate in Fig. 5(a).

The streaming latency is also a key factor affecting the VR video viewing experience. The saved percentage of the latency η_t for each scheme against the FC scheme is defined as $\eta_t = (t_f - t_s)/t_s \times 100\%$, where t_f and t_s are the latencies for the FC scheme and for the scheme to be compared, respectively. The curves of the saved latency versus the cache hit rate of the JERTC scheme and the OEC scheme are shown in Fig. 5(b). In the figure, when the cache hit rate of the scheme to be compared was more than 40%, the comparisons were performed with the result of FC at the cache hit rate of 40%. It is obvious that the proposed JERTC scheme can save more latency than the OEC scheme at the same cache hit rate. Averagely, the proposed scheme can save the latency by 10% over the OEC scheme and 80% over the FC scheme. What's more, the saved latencies of the both schemes grow with the increasing of the the cache hit rate because more VR video tiles were found in the cache nodes in the RAN and EPC.

3.4 Effects of capacity ratio and Zipf parameter on performance

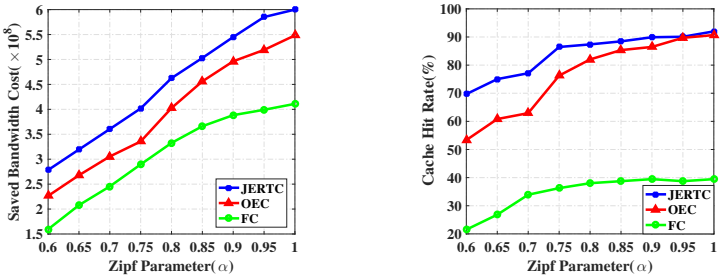


(a) (b)

Fig. 6. (a) The curves of the saved bandwidth cost vs. the capacity ratio and (b) the curves of the cache hit rate vs. the capacity ratio for the JERTC, OEC and FC schemes.

The capacity size affects the cache performance directly. The saved bandwidth and the cache hit rate were measured with a set of capacity ratios varying from 20% to 80% as shown in Fig. 6. In the experiments, the request routing followed the description in the second paragraph in Section 2. It can be seen from Fig. 6 that all three schemes can save more bandwidth and achieve higher cache hit rate with the increasing of the capacity ratio. It illustrates that larger capacity size will significantly increase the cache hit rate and correspondingly save more bandwidth cost. In Fig. 6(a), the proposed JERTC scheme can save the

most bandwidth cost among all three schemes. It indicates that the tile caching scheme can increase the cache hit rate of tiles due to its smaller cache size to cater for the viewport-requesting way against the full-size caching. This is also verified by the cache hit rate to capacity ratio comparisons among the JERTC, OEC and FC schemes, as shown in Fig. 6(b).



(a) (b)

Fig. 7. (a) The curves of the saved bandwidth cost vs. the Zipf parameter and (b) the curves of the cache hit rate vs. the Zipf parameter for the JERTC, OEC and FC scheme.

Zipf distribution parameter α also affects the performance of the caching schemes. It can be seen from Fig. 7(a) that the proposed JERTC scheme can save more bandwidth cost than the OEC and FC schemes with the increasing α. It is because larger α value increases the hit-rate of viewport requesting for each VR video in the caches for the JERTC scheme. It is finally evidenced by the increased cache bit-rate, as shown in Fig. 7(b). Though the other two schemes both improve the caching performance in bandwidth cost and cache hit rate with the increasing α, their improvements are smaller than that of the JERTC scheme due to the farther caching position for OEC scheme and the larger spatial caching size for FC scheme.

4 Conclusion

In this paper, a joint EPC and RAN tile-caching scheme of 360-degree VR videos is proposed for mobile networks. By fully considering the tiling characteristics of VR videos and the restriction nature of the cache space in mobile networks, 360-degree VR video tiles are jointly cached in both EPC and RAN using the 0-1 knapsack optimization. Experimental results show that the proposed joint EPC and RAN tile-caching scheme can significantly reduce the duplicate video tile transmissions which relieves the pressure on mobile networks and at the same time reduces the latency to ensure the requirements of VR applications. In our future work, a network-adaptive data scheduling will be studied and integrated with the scheme to further improve the VR video streaming performance.

References

1. Ahlehagh, H., Dey, S.: Video-aware scheduling and caching in the radio access network. IEEE/ACM Trans. Networking **22**(5), 1444–1462 (Oct 2014)
2. Andre, J., Siarry, P., Dognon, T.: An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization. Adv. Eng. Softw. **32**(1), 49–60 (Dec 2000)

3. Boyce, J., Alshina, E., Abbas, A., Ye, Y.: Jvet-d1030 r1: Jvet common test conditions and evaluation procedures for 360 video (Oct 2016)
4. Budagavi, M., Furton, J., Jin, G., Saxena, A. et al.: 360 degrees video coding using region adaptive smoothing. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 750–754 (Sept 2015)
5. Corbillon, X., Simon, G., Devlic, A., Chakareski, J.: Viewport-adaptive navigable 360-degree video delivery. In: 2017 IEEE International Conference on Communications (ICC). pp. 1–7 (May 2017)
6. Franky, O.E.A., Perdana, D., Negara, R.M., Sanjoyo, D.D. et al.: System design, implementation and analysis video cache on internet service provider. In: 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA). pp. 157–162 (July 2016)
7. Gaddam, V.R., Riegler, M., Eg, R., Griwodz, C. et al.: Tiling in interactive panoramic video: Approaches and evaluation. IEEE Trans. Multimedia **18**(9), 1819–1831 (Sept 2016)
8. Guntur, R., Ooi, W.T.: On tile assignment for region-of-interest video streaming in a wireless lan. In: Proceedings of the 22nd International Workshop on Network and Operating System Support for Digital Audio and Video. pp. 59–64. ACM (2012)
9. Hosseini, M., Swaminathan, V.: Adaptive 360 vr video streaming based on mpeg-dash srd. In: 2016 IEEE International Symposium on Multimedia (ISM). pp. 407–408 (Dec 2016)
10. Lim, S.Y., Seok, J.M., Seo, J., Kim, T.G.: Tiled panoramic video transmission system based on mpeg-dash. In: 2015 International Conference on Information and Communication Technology Convergence (ICTC). pp. 719–721 (Oct 2015)
11. Ohl, S., Willert, M., Staadt, O.: Latency in distributed acquisition and rendering for telepresence systems. IEEE Trans. Visual. Comput. Graphics **21**(12), 1442–1448 (Dec 2015)
12. Shen, S., Akella, A.: An information-aware qoe-centric mobile video cache. In: Proceedings of the 19th annual international conference on mobile computing & networking. pp. 401–412. ACM (2013)
13. Sitzmann, V., Serrano, A., Pavel, A., Agrawala, M. et al.: Saliency in vr: How do people explore virtual environments? IEEE Trans. Visual. Comput. Graphics **24**(4), 1633–1642 (April 2018)
14. Skupin, R., Sanchez, Y., Hellge, C., Schierl, T.: Tile based hevc video for head mounted displays. In: 2016 IEEE International Symposium on Multimedia (ISM). pp. 399–400 (Dec 2016)
15. Wang, X., Chen, M., Taleb, T., Ksentini, A. et al.: Cache in the air: exploiting content caching and delivery techniques for 5g systems. IEEE Commun. Mag. **52**(2), 131–139 (Feb 2014)
16. Xie, G., Li, Z., Kaafar, M.A., Wu, Q.: Access types effect on internet video services and its implications on cdn caching. IEEE Trans. Circuits Syst. Video Technol. **28**(5), 1183–1196 (May 2018)
17. Ye, Z., Pellegrini, F.D., El-Azouzi, R., Maggi, L. et al.: Quality-aware dash video caching schemes at mobile edge. In: 2017 29th International Teletraffic Congress (ITC 29). vol. 1, pp. 205–213 (Sept 2017)
18. Zhou, X., Sun, M., Wang, Y., Wu, X.: A new qoe-driven video cache allocation scheme for mobile cloud server. In: 2015 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE). pp. 122–126 (Aug 2015)