

Tile-Based Joint Caching and Delivery of 360° Videos in Heterogeneous Networks

Pantelis Maniotis
CSEE Department
University of Essex
Colchester, United Kingdom
p.maniotis@essex.ac.uk

Eirina Bourtsoulatze
EEE Department
University College London
London, United Kingdom
e.bourtsoulatze@ucl.ac.uk

Nikolaos Thomos
CSEE Department
University of Essex
Colchester, United Kingdom
nthomos@essex.ac.uk

Abstract—The recent surge of applications involving the use of 360° video challenges mobile networks infrastructure, as 360° video files are of significant size, and current delivery and edge caching architectures are unable to guarantee their timely delivery. In this paper, we investigate the problem of joint collaborative content-aware caching and delivery of 360° videos. The proposed scheme takes advantage of 360° video encoding in multiple tiles and layers to make fine-grained decisions regarding which tiles to cache in each small base station (SBS), and from where to deliver them to the end users, as users may reside in the coverage area of multiple SBSs. This permits to cache the most popular tiles in the SBSs, while the remaining tiles may be obtained through the backhaul. In addition, we explicitly consider the time delivery constraints to ensure continuous video playback. We evaluate the performance of the proposed method for various system parameters and compare it with schemes that do not consider 360° video encoding into multiple tiles and quality layers. The results make clear the benefits coming from caching and delivery decisions on per tile basis.

Index Terms—Collaborative caching, 360° video, tile encoding, layered video, distortion optimization.

I. INTRODUCTION

Nowadays we are witnessing an enormous increase in the traffic of 360° visual content originating from Virtual Reality (VR) and Augmented Reality (AR) applications. This growth is fuelled by the proliferation of devices that can display VR content, e.g. smartphones, tablets, Head Mounted Displays (HMDs) as well as services that offer users immersive multi-media experience such as online gaming.

A video frame in a 360° video encodes a 360° field of view. At any given time, a user views a portion of the scene, known as *viewport*. The displayed part of the scene may change according to the movement of the user's headset, thus, offering the user an immersive experience. This, however, comes at the cost of very high delivery bandwidth requirements, as the bitrate required to support a high Quality of Experience (QoE) is much larger than that for a regular video. Under these bandwidth requirements, the delivery of 360° video to wireless users in a cellular network given the strict delivery

deadlines imposed by VR and AR applications, becomes a very challenging problem.

To facilitate the delivery of massive video content in cellular networks, mobile network operators may enable edge caching [1], [2]. In edge caching systems, popular content is placed in the cache of the SBSs enabling to serve user requests locally. This limits the usage of the pricey backhaul links as well as reduces the latency through the use of short range communication, which improves the users' QoE. However, despite the existence of edge caching solutions for video [3]–[5], they are not readily deployable for the caching of 360° video files. This is due to the fact that 360° video frames encode the full 360° scene and require significant storage resources; however, only a small portion of the available scene, i.e., a viewport, is viewed by the user at any given time. This creates an additional degree of freedom in deciding which parts of the video file should be cached in the SBSs, as users are only interested in displaying a part of the scene and the delivery of the entire video may be unnecessary as parts of it will never be displayed. Without explicit consideration of these aspects of 360° video content and given the high-rate low-latency constraints for the delivery of 360° video, the growing bandwidth and caching requirements will put more pressure on mobile network operators' infrastructure, who will struggle to accommodate users' (often diverse) demands for 360° video content. Clearly, there is a need for novel methods to optimize the use of the available bandwidth and storage resources in order to support 360° video delivery and maintain high the users' QoE.

In this work, we propose a content-aware 360° video caching and delivery scheme for wireless cellular networks, that aims at maximizing the quality of the video delivered to the client population while respecting the delivery deadlines of the 360° video content. To this end, we leverage the benefits of encoding a 360° video into a number of independent tiles and quality layers as illustrated in Fig. 1. This approach allows us to deal with the diverse demands of the users in terms of the viewports they are interested in watching for each 360° video in a fine-grained way. Encoding into tiles and quality layers creates a flexible structure and enables collaborative caching of the various video tiles. Furthermore, layered coding allows to deliver only the viewport of interest in high quality, with the

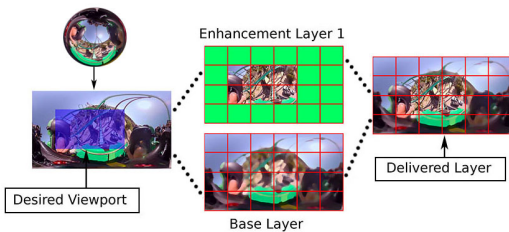


Fig. 1. Encoding of 360° video in two layers and multiple tiles.

rest of the viewports being delivered in lower quality (see Fig. 1). This permits to avoid rapid degradation of the perceived QoE in case of erroneous viewport prediction, while saving the network bandwidth resources. Evaluation of our method shows the advantages of the proposed approach compared to its counterparts, which do not consider tile encoding and/or cache collaboration. To the best of our knowledge, this is the first work that considers collaborative caching on per-tile basis for 360° videos.

II. RELATED WORK

Edge caching has been shown to improve the performance of cellular networks in terms of the observed delivery delay [3]–[5], energy consumption [6], and operating cost [7]. Users’ QoE can be improved by taking into account users’ association with more than one SBSs [8] and by allowing collaboration between the SBSs [3], [4]. SBSs collaboration accelerates content retrieval and reduces the usage of expensive backhaul links. The benefits of collaborative edge caching for video streaming have been studied in [9], where it is shown that the overall cache hit ratio and the QoE perceived by the users are increased when SBSs decide collaboratively which video files to cache. The QoE in the context of edge caching can be further improved by exploiting video encoding into multiple layers [5], which can be done by scalable coders.

The delivery of 360° video has been studied in [10], where the objective is to design multiple VR streams of different view ranges and find the optimal head angle-to-stream mapping under the bandwidth and storage constraints. Differently from [10], the estimation of encoding ladders in adaptive streaming systems for 360° videos has been examined in [11], where both the provider’s and the client’s perspectives are taken into account. A viewport-adaptive navigable 360° video communication system has been presented in [12]. This work suggests that sending full 360° video files to the users from where they can extract the viewport of interest leads to waste of bandwidth. On the other hand, sending only the demanded part of the viewport may lead to unacceptable delays. The solution proposed in [12] is a viewport-adaptive DASH-compatible scheme, where video representations are offered not only in multiple rates, but also with different quality of emphasized regions.

The encoding of a 360° video file into tiles for the purpose of streaming has been investigated in [13], [14]. This approach has shown significant performance improvement in streaming

systems without the use of edge caching. These streaming systems take into account that users are interested in viewing only a viewport of the 360° video scene, and hence there is no need to deliver the whole scene in high quality. The selection of the tiling scheme is investigated in [15], where the authors formulate the selection of the tiling scheme as an integer linear programming problem, where the optimal solution may be a non-uniform tiling scheme. Similarly to [15], the authors in [16] examine the impact of the selection of the tiling scheme on the compression and the resulting streaming bitrate. Differently from [15], [16], in [17] the authors investigate the performance of tile-based video streaming in 4G networks with respect to the coding efficiency and the bandwidth savings. The use of DASH for 360° videos is investigated in [18], where authors propose a video streaming scheme that aims to optimize the rate at which each tile is downloaded.

III. SYSTEM SETUP

We consider a mobile network, which consists of a set $\mathcal{N} = \{n_1, \dots, n_i, \dots, n_N\}$ of N small-cell base stations (SBS) and a macro-cell base station (MBS) denoted as n_{N+1} . The SBSs can communicate with the MBS and retrieve the requested 360° video from distant servers via backhaul links. For notational convenience, we define the set $\mathcal{N}_B = \mathcal{N} \cup \{n_{N+1}\}$ which includes all base stations of the cell. Each SBS n_i is equipped with a cache with capacity $C_i \geq 0$ (in Mbits), where content files can be cached.

Each user $u_i \in \mathcal{U} = \{u_1, \dots, u_i, \dots, u_U\}$ of the mobile network is associated with one or more SBSs depending on the transmission range of each SBS. The overlap in the coverage areas of the SBSs allows users to access multiple SBSs. We introduce the binary variable $\alpha_{nu} \in \{0, 1\}$ that equals to 1 when the user u resides in the coverage area of the SBS n , and 0, otherwise. The transmission range of the MBS is considered to be large enough so that each SBS can establish a connection with the MBS. User’s request can be satisfied by any of the associated SBSs, if the requested 360° video file is stored in their cache. Otherwise, the requested content is fetched from a remote server through the backhaul link with the help of the MBS. We define d_{nu} (in sec/Mbit) as the time needed to transmit 1Mbit of data from the n th SBS to the user u , and $d_{(N+1)u}$ as the corresponding time when the data is fetched from the backhaul. Obviously, $d_{(N+1)u} > d_{nu}$, $n \in \mathcal{N}$ due to the additional time needed to transmit the requested data from the backhaul to the MBS.

We assume that content retrieval is carried out in two phases, namely content placement, and content delivery. In the content placement phase, content is fetched during off-peak hours from distant back-end servers to the caches of the SBSs. Such offline caching can help to avoid network congestion during peak hours, as user requests are diverted from the remote content servers to the local caches (SBSs). In the delivery phase, the cached content is delivered to the users by either the SBSs or through the backhaul link via the MBS.

We denote the set of 360° videos that form the video content catalogue as $\mathcal{V} = \{v_1, \dots, v_i, \dots, v_V\}$. All video files are

stored at back-end content servers, while SBSs cache only part of the available content catalogue. Each 360° video file is encoded in a set of Groups of Pictures (GoP) \mathcal{G} . Each GoP consists of a number of independently coded tiles, which form the set \mathcal{T} . We assume that all 360° video files are encoded in the same number of tiles. Each tile is further encoded into a set of quality layers \mathcal{L} consisting of a base layer and multiple enhancement layers. The base layer offers a reconstruction of a tile at the lowest available quality, while enhancement layers gradually improve the reconstruction quality of each tile. For notational convenience, we will refer to tile t of the l th layer that belongs to the g th GoP of the v th video file as $vglt$. Finally, we denote the size of the tile $vglt$ (in Mbits) as o_{vglt} .

We introduce the auxiliary binary variable $z_{vglt}^u \in \{0, 1\}$ to denote the user requests, i.e., $z_{vglt}^u = 1$ when user u requests tile $vglt$, and $z_{vglt}^u = 0$, otherwise. To satisfy a user request for a viewport of a 360° video file at a specific quality, the user has to acquire all the tiles that cover the area of the viewport of that video at the desired quality. However, users may be interested in watching different parts of a video scene, while they may also request different video files. As a result, the popularity of tiles depends on both video and viewport popularity. In particular, the overlap between the various viewports shapes the popularity of each tile. Encoding of the 360° video files in layers and tiles offers greater flexibility in deciding which data should be stored in the SBSs caches.

IV. PROBLEM FORMULATION

In this paper, we aim at finding the optimal joint caching and delivery policy for tile-encoded 360° layered videos that maximizes the cumulative distortion reduction experienced by the users.

Let us introduce the binary variable $y_{vglt}^{nu} \in \{0, 1\}$, where $y_{vglt}^{nu} = 1$, if the tile $vglt$ will be delivered by base station $n \in \mathcal{N}_B$ to the user u , and $y_{vglt}^{nu} = 0$ otherwise. The routing decisions in our system can be described by the vector:

$$\mathbf{y} = (y_{vglt}^{nu} \in \{0, 1\} : n \in \mathcal{N}_B, u \in \mathcal{U}, vgl \in \mathcal{V} \times \mathcal{G} \times \mathcal{L} \times \mathcal{T}) \quad (1)$$

Similarly, we define the binary decision variable $x_{vglt}^n \in \{0, 1\}$, which takes the value 1 when the tile $vglt$ is cached at the n th SBS, and 0, otherwise. Hence, the caching decisions for the entire system are described by the vector:

$$\mathbf{x} = (x_{vglt}^n \in \{0, 1\} : n \in \mathcal{N}, vgl \in \mathcal{V} \times \mathcal{G} \times \mathcal{L} \times \mathcal{T}) \quad (2)$$

When a user obtains a tile, the distortion observed by the user decreases. Let us denote by δ_{vglt} the average distortion reduction associated with tile $vglt$. The optimization objective is to maximize the cumulative distortion reduction over the users population. Denoting this quantity by D , we have:

$$D = \sum_{n \in \mathcal{N}_B} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} \sum_{g \in \mathcal{G}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} \delta_{vglt} y_{vglt}^{nu} \quad (3)$$

Thus, the problem of joint caching and delivery of 360° videos can be formally expressed as:

$$\mathcal{P} : \max_{\mathbf{x}, \mathbf{y}} D \quad s.t. : \quad (4)$$

$$\sum_{v \in \mathcal{V}} \sum_{g \in \mathcal{G}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} o_{vglt} x_{vglt}^n \leq C_n, \forall n \in \mathcal{N} \quad (5)$$

$$y_{vglt}^{nu} \leq z_{vglt}^u \alpha_{nu} x_{vglt}^n, \quad \forall n \in \mathcal{N}, u \in \mathcal{U}, vgl \in \mathcal{V} \times \mathcal{G} \times \mathcal{L} \times \mathcal{T} \quad (6)$$

$$\sum_{n \in \mathcal{N}_B} y_{vglt}^{nu} \leq 1, \forall u \in \mathcal{U}, vgl \in \mathcal{V} \times \mathcal{G} \times \mathcal{L} \times \mathcal{T} \quad (7)$$

$$\sum_{n \in \mathcal{N}_B} y_{vg(l+1)t}^{nu} \leq \sum_{n \in \mathcal{N}_B} y_{vglt}^{nu}, \quad \forall u \in \mathcal{U}, vgl \in \mathcal{V} \times \mathcal{G} \times \mathcal{L} \times \mathcal{T} \quad (8)$$

$$\sum_{n \in \mathcal{N}_B} \sum_{g' \in \{1, \dots, g\}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} o_{vg'lt} d_{nu} y_{vg'lt}^{nu} \leq t_{app} + (g-1)t_{disp}, \forall u \in \mathcal{U}, v \in \mathcal{V}, g \in \mathcal{G} \quad (9)$$

Eq. (5) is the cache capacity constraint. Constraint (6) captures the fact that in order to deliver a tile requested by a user from the cache of an SBS, the tile has to be stored in the cache and the user has to reside in the coverage area of the SBS. Constraint (7) ensures that each tile will be received only once by each client. In addition, we have to take into consideration limitations arising from the encoding in multiple tiles and layers. In order to achieve the quality that corresponds to the l th layer, the user should receive not only the l th layer, but also all previous layers. This constraint is captured by (8). Inequality (9) corresponds to the requirement for smooth playback, where t_{app} and t_{disp} correspond to the playback delay and the time needed to display a GoP, respectively.

V. DISTRIBUTED ALGORITHM

The optimization problem in Eq. (4) is NP-hard. This can be shown by decomposing it into a number of subproblems, where each subproblem consists of two NP-hard components. To solve the problem efficiently, we simplify the original problem by introducing a fairness constraint with respect to the cache space allocation and the delivery delay per GoP. Specifically, we limit the cache available for each GoP to be at least $\lfloor C_n/G \rfloor$ and obtain a modified cache capacity constraint:

$$\sum_{v \in \mathcal{V}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} o_{vglt} x_{vglt}^n \leq C_n(g), \forall n \in \mathcal{N} \quad (10)$$

$C_n(g)$ stands for the cache space available in n th SBS for the g th GoP and is defined as: $C_n(g) = \lfloor C_n/G + C_n^{rem}(g-1) \rfloor$, where $C_n^{rem}(g) = C_n(g) - \sum_{v \in \mathcal{V}, l \in \mathcal{L}, t \in \mathcal{T}} o_{v(g)lt} x_{v(g)lt}^n$ corresponds to the amount of cache that has not been filled in with content after making the caching decisions for g th GoP. Similarly, we assume that the delay for delivering each GoP is $t_{app}/G + t_{disp}$ and rewrite the delivery delay constraint as:

$$\sum_{n \in \mathcal{N}_B} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} o_{vglt} d_{nu} y_{vglt}^{nu} \leq t(g), \forall u \in \mathcal{U}, v \in \mathcal{V} \quad (11)$$

where $t(g) = t_{app}/G + t_{disp} + t^{rem}(g-1)$ represents the delivery delay of the g th GoP and $t^{rem}(g) = t(g) - \sum_{n \in \mathcal{N}_B} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} o_{v(g)lt} d_{nu} y_{v(g)lt}^{nu}$ is the time remaining from the delivery of g th GoP.

The introduction of the above two fairness constraints allows to decompose the problem in (4) into G subproblems. We can rewrite Eq. (4) as $D = \sum_g D_g$ where

$$D_g = \sum_{n \in \mathcal{N}_B} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} \delta_{vglt} y_{vglt}^{nu} \quad (12)$$

By replacing the coupling cache and delay constraints in Eqs. (5) and (9) by the fairness constraints in Eqs. (10) and (11), we obtain the following G subproblems:

$$\mathcal{P}_g : \max_{\mathbf{x}_g, \mathbf{y}_g} D_g \quad \text{s.t.: (6), (7), (8), (10) and (11)} \quad (13)$$

where \mathbf{x}_g and \mathbf{y}_g correspond to the cache and routing variables of the g th GoP.

The global routing and caching policies \mathbf{x} and \mathbf{y} are obtained after solving the subproblems in (13) sequentially for each GoP starting with the first GoP. To solve each subproblem \mathcal{P}_g , we use the method of Lagrange partial relaxation. We relax the constraint (6), which enables us to decompose each subproblem \mathcal{P}_g into two components: one that involves only the cache related variables and another that involves only the routing variables. Given the set of Lagrange multipliers for every $g \in \mathcal{G}$:

$$\lambda_g = (\lambda_{vgt}^{nu} \geq 0, \forall n \in \mathcal{N}, u \in \mathcal{U}, v \in \mathcal{V}, l \in \mathcal{L}, t \in \mathcal{T}) \quad (14)$$

we obtain the Lagrangian function:

$$\begin{aligned} L(\lambda_g, \mathbf{x}_g, \mathbf{y}_g) = \max_{\mathbf{x}_g, \mathbf{y}_g} & \left[\sum_{n \in \mathcal{N}} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} (\delta_{vgt} \right. \\ & - \lambda_{vgt}^{nu}) y_{vgt}^{nu} + \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} \delta_{vgt} y_{vgt}^{(N+1)u} \\ & \left. + \sum_{n \in \mathcal{N}} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} \lambda_{vgt}^{nu} z_{vgt}^u \alpha_{nu} x_{vgt}^n \right] \end{aligned} \quad (15)$$

and the Lagrangian dual problem:

$$\min_{\lambda_g \geq 0} L(\lambda_g, \mathbf{x}_g, \mathbf{y}_g) \quad (16)$$

We can easily observe that the Lagrange function consists of two independent optimization sub-subproblems $P1_g$ and $P2_g$, where:

$$P1_g : \max_{\mathbf{x}_g} \left[\sum_{n \in \mathcal{N}} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} \lambda_{vgt}^{nu} z_{vgt}^u \alpha_{nu} x_{vgt}^n \right] \quad (17)$$

s.t.: (10)

and

$$\begin{aligned} P2_g : \max_{\mathbf{y}_g} & \left[\sum_{n \in \mathcal{N}} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} (\delta_{vgt} \right. \\ & - \lambda_{vgt}^{nu}) y_{vgt}^{nu} + \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} \delta_{vgt} y_{vgt}^{(N+1)u} \left. \right] \end{aligned} \quad (18)$$

s.t.: (7), (8) and (11)

Since $P1_g$ involves only the caching variables \mathbf{x}_g of the subproblem \mathcal{P}_g , we refer to it as the *caching component*. In order to determine the optimal cache allocation \mathbf{x}_g for $P1_g$, the problem is further decomposed into N 0-1 knapsack problems, which can be solved independently using optimization methods such as Dynamic Programming [19]. In this case, we maximize the objective function of $P1_g$, considering that the cache space of each SBS is a knapsack. The optimization subproblem $P2_g$ involves only the routing decision variables \mathbf{y}_g , hence it is called the *routing component*, and is solved with the help of CPLEX [20].

To solve the Lagrangian dual problem in (16), we apply the sub-gradient method to update the dual variables iteratively.

We start with non-negative values for the Lagrangian multiplier variables, and then based on the solution of the $P1_g$ and $P2_g$, in each iteration τ the dual variables are updated as follows:

$$\begin{aligned} \lambda_{vgt}^{nu}(\tau + 1) = & [\lambda_{vgt}^{nu}(\tau) \\ & - \sigma(\tau)(-y_{vgt}^{nu}(\tau) + z_{vgt}^u \delta_{vgt} \alpha_{nu} x_{vgt}^n(\tau))]^+ \end{aligned} \quad (19)$$

where $[s]^+ = \max(0, s)$ and $\sigma(\tau)$ expresses the step size which controls the convergence properties of the sub-gradient algorithm at each iteration. If we denote by LB the lower bound of the subproblem \mathcal{P}_g , UB the value of the Lagrange function at iteration τ , and $\phi(\tau)$ the subgradient, the step size can be easily calculated using the formula below:

$$\sigma(\tau) = w \frac{UB - LB}{\|\phi(\tau)\|^2} \quad (20)$$

where $w \in (0, 2]$ is a positive constant that scales the step size. The value of UB is the value of the Lagrange function at each iteration, while LB equals to the value of the objective function of the subproblem \mathcal{P}_g at a feasible solution.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed algorithm exploiting cache collaboration and coding in multiple layers and tiles. Due to space limitations, we only showcase the impact of the *cache size*, the *SBS communication link delay*, the *video popularity distribution* and the *density of the SBSs network*, and refer the interested readers to [21] for a more comprehensive study, including the impact of the backhaul link delay and the viewport popularity distribution.

We compare our method with the following benchmark schemes:

- *Independent Caching-No Tiles (ICNT)*: each video is encoded in a single quality layer and tile and each user is associated only with one SBS. The caching and delivery policy is found by solving (4) for each SBS separately.
- *Joint Caching-No Tiles (JCNT)*: each video is encoded in a single quality layer and tile, but contrary to ICNT, each user is associated with all SBSs whose coverage area it resides in. The caching and routing decisions are made jointly.

Unless otherwise stated, we consider a cellular network with 5 SBSs and 30 users randomly placed in the macrocell. The coverage radius of SBSs is set to 200m, while each SBS has cache capacity to store 10% of the size of the video catalogue. The transmission delay on links connecting SBSs to users is set to 1s/Mbit, while the transmission delay for fetching the content from the backhaul to any user is 5s/Mbit

The content library contains 10 videos with resolution 1280×720 . We assume that 4 of the videos have characteristics similar to ‘‘Hog Rider’’, 3 similar to ‘‘Roller Coaster’’ and 3 similar to ‘‘Chariot Race’’.¹ The videos are encoded into 12 tiles per frame and 2 quality layers. The average size and the distortion reduction per tile and layer are given in Table I. We assume that the size of each viewport is 2×2 tiles. Each video consists of $G = 30$ GoPs with duration of 1s per GoP. The

¹These videos were obtained from YouTube.

TABLE I
AVERAGE SIZE (IN MBITS) AND DISTORTION REDUCTION PER TILE AND LAYER FOR THE CONSIDERED VIDEO SEQUENCES

Video	O_{vg1t}	O_{vg2t}	δ_{vg1t}	δ_{vg2t}
“Hog Ride”	0.010	0.125	118	125
“Roller Coaster”	0.016	0.167	292	298
“Chariot Racer”	0.029	0.275	187	192

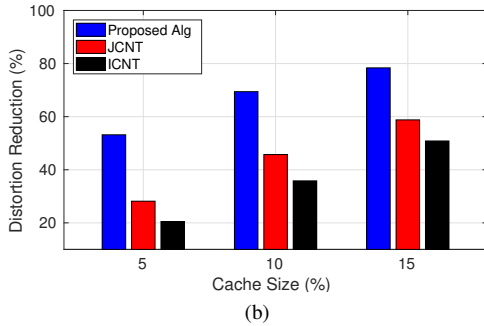
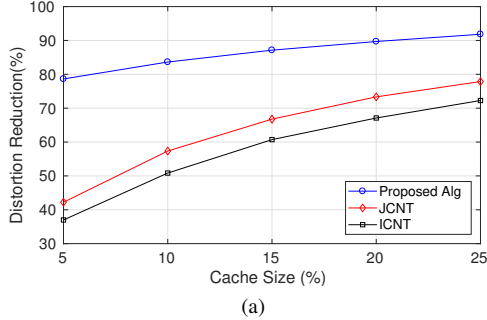


Fig. 2. (a) Distortion reduction and (b) cache hit ratio with respect to the cache size for all schemes under comparison.

playback delay for each video is 1s, and the display time of a GoP is 1s. The videos are encoded using the scalable extension (SHVC) of HEVC/H.265 standard, which allows encoding in tiles and layers.

The probability p_v that video v is requested by a user follows Zipfian distribution with shape parameter $\eta = 1$:

$$p_v = \frac{1/v^\eta}{\sum_{v \in \mathcal{V}} 1/v^\eta}.$$

We assume that all users demand the base layer of the whole video of their preference (this provides a minimum reconstruction quality for all tiles) and the enhancement layer only for the tiles that form the requested viewport. To study the effect of non-uniform popularity of tiles we consider that each 360° video is available in six different viewports without allowing viewports to fold around. Assuming that the requests for these viewports are uniform, results in central tiles (which belong to multiple viewports) having higher popularity.

1) *Cache Size*: Fig. 2a shows the impact of the cache size of SBSs on the achieved distortion reduction D expressed as the percent of the maximum achievable cumulative distortion reduction. We can observe that our proposed scheme outperforms the JCNT and ICNT schemes by a large margin, especially for small cache sizes (in the region of 5%-10% of the

content library, which are typical values for networks handled by mobile network operators). This is due to the fact that our scheme not only exploits SBSs collaboration and makes joint caching and routing decisions (as in JCNT), but also takes advantage of video encoding in tiles and layers (which is not possible neither in ICNT nor in JCNT). This allows for granularity in caching and routing decisions, and permits to cache the most popular and significant chunks of data locally in the SBSs, while preserving high data diversity across the SBSs thanks to collaboration among SBSs. This translates into the majority of tiles being delivered through local short range communications with low delay, while the rest can be delivered via backhaul within the delivery deadlines. This is often not possible in JCNT and ICNT where encoding is done in one layer and tile, and the delay constraint may not allow timely delivery of entire large data chunks from remote locations.

Our observations are supported by Fig. 2b, which depicts the achieved cache hit ratio with respect to the cache size. High cache hit ratio achieved by our proposed scheme indicates that the majority of requests are served locally from the SBSs caches. On the contrary, the low cache hit ratio in JCNT and ICNT schemes suggests that user requests are not served locally and content is fetched from the remote servers.

2) *SBS communication link delay*: Fig. 3a illustrates the effect of the delay of the communication link between the users and the SBSs on the distortion reduction D . We can observe that the performance of the proposed algorithm is only slightly affected by the increase in the SBS delay. This is due to the fact that encoding in tiles and layers, which produces data chunks of small size, permits to deliver most of the requested tiles locally from SBSs within the application delivery deadlines. On the contrary, in ICNT and JCNT the increase in the SBSs link delay prevents the timely delivery of large data chunks resulting in significant decrease in the achieved distortion reduction.

3) *Video popularity distribution*: In Fig. 3b, we examine the effect of the skewness parameter of the Zipfian distribution on the distortion reduction. Clearly, an increase in the η value leads to higher distortion reduction for all the under-examination schemes. For small values of η , i.e., very diverse content demands, the difference between the proposed scheme and ICNT and JCNT is approximately 36% and 30%, respectively. This is due to the encoding in tiles which permits to cache the base layer tiles locally and hence satisfy most of the users requests with a basic video quality. This is not the case in ICNT and JCNT as without encoding in tiles and layers the local caches are occupied by large data chunks which can satisfy only a small percent of the diverse users' requests. The performance gap closes when η value becomes higher, as the majority of the user requests refer to a smaller number of videos. High η values are more beneficial for ICNT and JCNT schemes, as these schemes do not encode the video in tiles and layers, but even for such η values, ICNT and JCNT schemes performance remains inferior to the proposed scheme.

4) *Density of SBSs*: We explore how the density of SBSs impacts the performance of all schemes in Fig. 3c. To per-

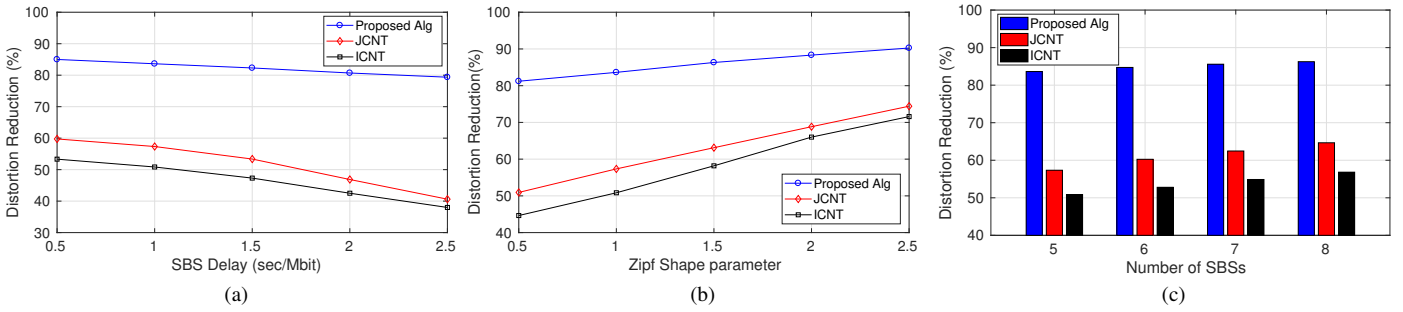


Fig. 3. Distortion reduction with respect to (a) the SBS delay, (b) the Zipf shape parameter and (c) the density of the SBSs for all schemes under comparison.

form this comparison, we add SBSs to the original network topology, by randomly placing them in the macrocell. We examine topologies with 5 up to 8 SBSs (recall that the original network setup had 5 SBSs). From the results, we can note that with more SBSs the distortion reduction improves for all the compared schemes and that the proposed scheme outperforms significantly ICNT and JCNT. This is attributed to the fact that more users are located in the overlap area of multiple SBSs. Hence, the proposed scheme and JCNT, which jointly decide the routing and caching policies, exploit the newly emerging caching opportunities. Also, we can see that the performance of ICNT improves as well. This is because higher SBSs density results in offloading more content to the SBSs, which limits the use of the backhaul link.

VII. CONCLUSION

In this work, we studied the problem of the 360° video delivery and caching in cellular networks comprising an MBS and multiple SBSs. To maximize the quality of the video delivered to the end users, we exploit advanced video coding tools that permit encoding of video in multiple tiles and layers offering greater granularity of information. We also exploit SBSs collaboration opportunities to prevent neighboring SBSs from caching the same video data. The proposed algorithm takes into consideration not only the content importance but also video popularity at tile level, to jointly decide the routing and caching policies. As the original problem is of high complexity, we decompose it into a number of subproblems and decouple the subproblems into their routing and caching components. The latter are solved using Lagrange decomposition. The experimental evaluation shows that collaborative caching and video encoding into tiles and quality layers allows to cache the most important data locally in the SBSs and deliver it in a timely manner to the users reducing the load of the backhaul links. Hence, our solution outperforms significantly methods that do not use collaboration and/or encoding into multiple tiles and quality layers.

REFERENCES

- [1] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: design aspects, challenges, and future directions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 22–28, Sept. 2016.
- [2] T. X. Vu, S. Chatzinotas, and B. Ottersten, "Edge-caching wireless networks: Performance analysis and optimization," *IEEE Trans. on Wireless Communications*, vol. 17, no. 4, pp. 2827–2839, Apr. 2018.
- [3] Q. Gong, J. W. Woods, K. Kar, and J. Chakareski, "Fine-grained scalable video caching," in *Proc. of IEEE Int. Symp. on Multimedia (ISM'15)*, Miami, FL, USA, Dec. 2015, pp. 101–106.
- [4] B. Zhang, Z. Liu, S. H. G. Chan, and G. Cheung, "Collaborative wireless freeview video streaming with network coding," *IEEE Trans. on Multimedia*, vol. 18, no. 3, pp. 521–536, Mar. 2016.
- [5] K. Poularakis, G. Iosifidis, A. Argyriou, I. Kotsopoulos, and L. Tassioulas, "Distributed caching algorithms in the realm of layered video streaming," *IEEE Trans. on Mobile Computing*, in press.
- [6] B. Perabathini, E. Baştuğ, M. Kountouris, M. Debbah, and A. Conte, "Caching at the edge: A green perspective for 5g networks," in *Proc. of IEEE ICC Workshops*, London, UK, Jun. 2015.
- [7] S. Zhang, N. Zhang, P. Yang, and X. Shen, "Cost-effective cache deployment in mobile heterogeneous networks," *IEEE Trans. on Vehicular Technology*, vol. 66, no. 12, pp. 11 264–11 276, Dec. 2017.
- [8] Y. Wang, X. Tao, X. Zhang, and G. Mao, "Joint caching placement and user association for minimizing user download delay," *IEEE Access*, vol. 4, pp. 8625–8633, 2016.
- [9] J. George and S. Sebastian, "Cooperative caching strategy for video streaming in mobile networks," in *Proc. of Int. Conf. on Emerging Technological Trends (ICETT'16)*, Kollam, India, Oct. 2016.
- [10] G. Cheung, Z. Liu, Z. Ma, and J. Z. G. Tan, "Multi-stream switching for interactive virtual reality video streaming," in *Proc. of IEEE Int. Conf. on Image Processing (ICIP'17)*, Beijing, China, Sep. 2017.
- [11] C. Ozcinar, A. D. Abreu, S. Knorr, and A. Smolic, "Estimation of optimal encoding ladders for tiled 360 vr video in adaptive streaming systems," in *Proc. of IEEE Int. Symp. on Multimedia (ISM'17)*, Taichung, Taiwan, Dec. 2017.
- [12] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. of IEEE ICC*, Paris, France, May 2017.
- [13] M. Hosseini, "View-aware tile-based adaptations in 360 virtual reality video streaming," in *Proc. of IEEE Int. Conf. on Virtual Reality (VR'17)*, Los Angeles, CA, USA, Mar. 2017.
- [14] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using mpeg-dash," in *Proc. of the 7th Int. Conf. on Multimedia Systems (MMSys'16)*, Klagenfurt, Austria, 2016.
- [15] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "Optile: Toward optimal tiling in 360-degree video streaming," in *Proc. of the ACM Int. Conf. on Multimedia*, ser. MM '17. New York, NY, USA: ACM, 2017.
- [16] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hecv-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proc. of ACM on Multimedia Conf. (ACM MM'16)*, Amsterdam, Netherlands, 2016.
- [17] W. C. Lo, C. L. Fan, S. C. Yen, and C. H. Hsu, "Performance measurements of 360 video streaming to head-mounted displays over live 4g cellular networks," in *Proc. of APNOMS'17*, Seoul, South Korea, Sep. 2017.
- [18] S. Rossi and L. Toni, "Navigation-aware adaptive streaming strategies for omnidirectional video," in *Proc. of IEEE 19th Int. Workshop on Multimedia Signal Processing (MMSp'17)*, Luton, UK, Oct. 2017.
- [19] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, 2004.
- [20] ILOG CPLEX optimization studio. <http://is.gd/3GGOfp>.
- [21] P. Maniotis, E. Boursoulatzte, and N. Thomos, "Tile-based joint caching and delivery of 360° videos in heterogeneous networks," *arXiv:1902.09581v1 [cs.NI]*, 2019.