Video Delivery over Heterogeneous Cellular Networks: Optimizing Cost and Performance

Konstantinos Poularakis, George Iosifidis, Antonios Argyriou, and Leandros Tassiulas

***Abstract*—Video delivery to mobile users is one of the largest challenges that network operators face today. In this work we consider a heterogeneous cellular network with storage capable small-cell base stations, and study this problem for pre-stored video files that can be encoded with two different schemes, namely versions or layers, in various qualities. We introduce a framework for the joint derivation of video caching and routing policies for users with different quality requirements. This allows the operator to optimize a balanced objective of incurred servicing cost, and users experienced delay, according to his priorities. The numerical results indicate that versions and layers may have different impact on the delay and servicing cost, depending on the diversity of users' demand, and that the cost-delay trade off is affected by the network's load.**

I. Introduction

Motivation. Nowadays there is a tremendous growth in the number of mobile users viewing videos [1], which are encoded and pre-stored on servers and delivered over cellular networks. Mobile network operators (MNOs) strive to serve these massive requests and achieve the minimum possible video delivery delay. This is very important since it is the main criteria for the users' perceived satisfaction. However, delivering this content puts unprecedented pressure on the networks and often yields a very high servicing cost for the operators. Achieving the right balance between this cost and the delivery delay experienced by the users is currently one of the most important challenges for the MNOs.

This problem becomes even more challenging today where many operators deploy 4G Heterogeneous Cellular Networks (HCNs). These are actually conventional cellular networks overlaid with small-cell base stations (SCBSs), such as pico-cells and femtocells, which are connected to the core network with capacitated backhaul links [2]. In HCNs, mobile users are concurrently in range with multiple base stations, and hence the operator can use multiple paths to route content to them. Moreover, the MNO can proactively cache at certain SCBSs popular video items [3], for which recurring requests are expected [4]. Field trials [5] have revealed that this technique improves the user experienced delay, and at the same time reduces the network servicing cost. Clearly, video delivery over HCNs raises unique technical challenges as there are many possible routing and caching policies.

Due to the heterogeneity of HCNs, these different policies may yield different network cost. The latter depends on several parameters, such as the base station load, which determines their energy consumption [6] and the bandwidth cost of the backhaul links [2]. For example, serving a user by a macrocellular base station induces to the operator higher cost than serving it with a low-power SCBS in close proximity [7], especially if the latter has the requested video already cached. On the other hand, these decisions impact the user perceived network performance (in terms of delay). For example, small cells introduce an additional hop in the routing path (i.e., the SCBS backhaul links) and hence, in some cases, significant delay [2]. As is typical in these scenarios, improving the network performance may increase the network cost, and hence the operator needs to carefully determine their balance according to his preferences.

This issue is further perplexed due to the particular characteristics of video delivery. Specifically, each video file should be available in various qualities since users often have different (minimum) quality requirements. To achieve this, every video can be encoded into multiple versions which differ in quality and rate (*versions*). Another option is scalable video coding (SVC) (*layers*) where each video is encoded into different layers which, when combined, produce a quality that increases as more layers are used. This technique introduces an encoding overhead but offers network flexibility since the layers of each file can be cached at different base stations and/or routed over different paths. The MNO can use versions, layers or a mixture of them for the video files.

Obviously, the HCN operators have a large repertoire of video encoding, caching and routing decisions for servicing the user requests. In order to achieve his balanced performance-cost objective, the operator has to jointly optimize these decisions. Clearly, this is an important problem that differs substantially from previous related studies for wired networks [8], or cellular networks [6] that did not consider backhaul-constrained and storage-capable SCBSs.

Contributions. In this work, we consider an HCN mobile operator (referred to as MNO) and study the problem of optimizing the servicing cost and the delivery delay for video requests of mobile users. We assume that the MNO is either a (self-contained) Telco-CDN or it cooperates closely with a CDN[1]. Therefore he entirely determines the video delivery policy which comprises the routing, caching and video encoding decisions. We study the system for a certain time period (e.g., several hours or few days) during which the users demand for a set of popular video files is assumed to be known in advance, as in [3], [9], [10]. This assumption is also motivated by recent measurement-based studies which indicated that a small number of video files often account for a large portion of traffic, especially for users located in certain areas (e.g., a university campus [4]), or embedded in a social

The authors are with the Department of Electrical and Computer Engineering, University of Thessaly, Greece. Email: kopoular, giosifid, anargyr, leandros @inf.uth.gr

[1]Such architectures are gaining increasing interest, e.g. see AKAMAI Press Release: Swisscom and Akamai Enter Into a Strategic Partnership, March 2013.

network [11].

We group users in *user classes*, based on their locations, and study average delay performance and cost metrics, for the users and the operator respectively. This allows us to optimize routing decisions jointly with caching decisions (which cannot be taken on a per-user basis). Besides, determining the servicing policy for each user independently (or, worse, for each request) would induce significant computational burden to the MNO. The video delivery policy is derived by the solution of a challenging optimization problem which includes the discrete caching decisions and a large constraint set for link capacities and cache sizes. The presented framework enables each operator to balance the cost-performance objective by tuning a simple balancing parameter.

Using system parameters driven from real traces datasets, we investigate numerically the impact of the video encoding decisions on the balanced delay and servicing cost. We find that when the user demand is homogeneous in terms of requested video quality, the operator can improve his balanced objective by using versions instead of layered encoding. However, as the users' demand becomes more diverse, layered encoding can be more beneficial, as it allows for more flexible caching and routing decisions. Moreover, we characterize the delay - cost tradeoff. We find that improving the delivery delay (by tuning properly the balancing parameter) by an average 10% may increase the servicing cost from 10% up to 30% depending on the load of the network (users' requests).

Summarizing, the contributions of this work are as follows:

- *Optimization Framework*. We introduce a framework for the joint optimization of video encoding, caching and routing decisions, that minimize a balanced objective of average delay and servicing cost. Our model considers realistic aspects of HCNs such as the capacitated backhaul links and the constraints for the cache sizes and the wireless capacity of the SCBSs.
- *Video Encoding Policies*. We explicitly model and study the impact of the employed video encoding scheme (versions or layers) on the servicing cost and delay. We explain under which conditions the operator should select one of them or even employ both of them. Also, we discuss how our analysis accounts for the video streaming model requirements.
- *Performance Evaluation*. Our study is generic which allows us to investigate the impact of several system parameters. Based on our numerical analysis, we conclude that video encoding decisions are affected by the homogeneity of users requests, the capacity of the SCBSs and the encoding overhead of layering. Also, we show that improving the delay up to 10% may induce additional servicing cost which can reach 30% when the network is heavily loaded.

The rest of the paper is organized as follows. Sec. II introduces the system model and the problem. In Sec. III we solve the video delivery problem for the case of versions encoding. We extend our methodology for the case that both versions and layers are used and discuss the implications for video streaming in Sec. IV. Sec. V provides performance evaluation results, Sec. VI reviews our contribution compared to related works, and we conclude in Sec. VII.

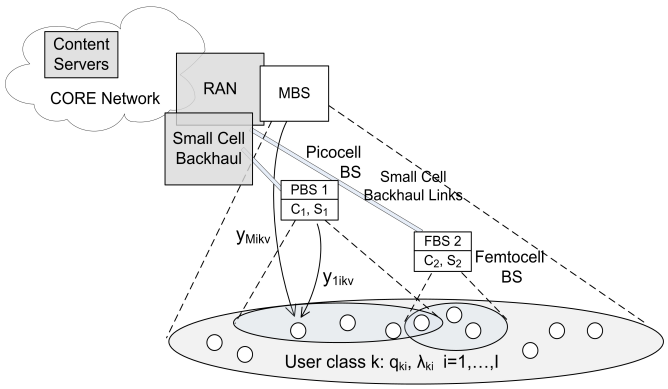

Fig. 1. A multi-tier cellular architecture with store-capable base stations.

II. System Model and Problem Statement

A. System Model

HCN Architecture. The system architecture is depicted in Fig. 1. We study the downlink operation of an HCN macrocell with one macrocellular BS (MBS), hereafter indexed M, a set (tier) $\mathcal{N}_\mathcal{P} \triangleq \{1, 2, \ldots, N_P\}$ of $|\mathcal{N}_\mathcal{P}|$ picocell base stations (PBSs) covering smaller areas within the macrocell, and a set $\mathcal{N}_\mathcal{F} \triangleq \{1, 2, \ldots, N_F\}$ of $|\mathcal{N}_\mathcal{F}|$ femtocell base stations (FBSs) with transmission range of few tens of meters[2]. We denote as $\mathcal{N} \triangleq \mathcal{N}_\mathcal{P} \cup \mathcal{N}_\mathcal{F}$ the set of all small cell base stations (SCBSs).

We study the system for a certain time period during which each SCBS $n \in \mathcal{N}$ has an average wireless capacity of $C_n \geq 0$ bps, while the capacity of the macrocell is $C_M \geq 0$ bps. BSs of the same type have similar characteristics but may differ in certain cases (e.g., due to location-dependent shadowing effects). The MBS coverage overlaps with all the other base stations, while it is also possible to have overlapping femto or picocell base stations [12]. We consider *disjoint subchannel allocation* among different tiers of BSs, which is one of the prevalent options for small cell deployment [13], and has improved performance especially for dense deployments [14]. Neighboring BSs in the same tier can also be assigned orthogonal frequency bands or employ enhanced inter-cell interference coordination techniques[3] (eICIC) proposed in LTE Rel. 10.

Each SCBS $n \in \mathcal{N}$ is connected to the core network through a wired or wireless backhaul link of average capacity $G_n \geq 0$ bps. These links connect the SCBS to certain aggregation points, e.g., fibre cabinets close to macrosites (see [2] for a survey on this). The MBS is connected to the core network through the typical high-capacity RAN backhaul. Finally, each SCBS n is endowed with a certain storage capacity of $S_n \geq 0$ bytes.

Video Encoding and Multiple Qualities. We study the delivery of a large set $\mathcal{I} \triangleq \{1, 2, \ldots, I\}$ of video files, each one of which can be delivered in $Q > 1$ different quality

[2]The analysis can be extended for more classes of small cell base stations (e.g., microcells), and can be generalized for multiple macrocells.

[3]The implementation of such schemes requires signaling between BSs so that they optimally adjust the transmission power in different OFDMA subcarriers [15], and properly allocate the frequency-time slots (bearers).

levels. The term quality level in this paper can correspond to different spatial resolutions (frame sizes), different temporal resolutions (frame rates), or different SNR qualities (controlled at the video coder). We assume that there is a set $\mathcal{V}$ of versions that can be offered for each file $i \in \mathcal{I}$. Each version $v \in \mathcal{V}$ corresponds to a certain quality level (it is $|\mathcal{V}| = Q$) and has size o_{iv} bytes which increases with the quality, i.e., $o_{iv} \geq o_{iu}$ if $v > u$ (assuming there is an ordering in versions wrt their quality and size).

Also, we assume that there is a set of layers $\mathcal{L}$ that can be offered for each video file when it is encoded with scalable video coding (SVC). This is an extension of the H.264/MPEG4-AVC standard that offers, among others, quality scalability [16]. A video decoder can reconstruct the video sequence by receiving a subset of them. In order to decode the layer l, all preceding layers $l' \leq l$ of the same video file should be available[4]. Let o_{il} denote the size of layer $l \in \mathcal{L}$ of file i. Compared to versions, layered encoding typically incurs an encoding overhead:

$$\sum_{l=1}^{q} o_{il} = o_{iv}(1 + R_i^q),\ v = q,\ \forall q \in \{1, 2, ..., Q\} \quad (1)$$

where $R_i^q > 0$ is the encoding overhead for the quality level q of file i that can be calculated with experimental methods [17]. Layered encoding through SVC has been used extensively the last few years in the real-world video-conferencing systems[5].

User Requests. We model demand by introducing a set $\mathcal{K} \triangleq \{1, 2, \ldots, K\}$ of user classes, each one representing a subset of users in the same location (very small subregion of the macrocell), asking for (possibly) different files with certain minimum quality requirements[6]. Video files are delivered through the streaming mechanism. This means that the user starts decoding and rendering the video file before it receives it in its entirety. This aspect is analyzed in detail in Section IV.B. User locations can be random, e.g., following a uniform distribution. Let $\lambda_{ki} \geq 0$ denote the demand (i.e., number of requests) of user class k for file $i \in \mathcal{I}$ with minimum quality $q_{ki} > 0$. This means that user class k should get either one version v with $v \geq q_{ki}$, or all layers up to q_{ki}, i.e. $l = 1, 2, \ldots, q_{ki}$. Unless otherwise specified, a user requesting (minimum) quality q_{ki} can be served with higher quality as well. The total demand that must be served by the MNO for the specific macrocell is:

$$\Lambda = \big((\lambda_{ki},\, q_{ki}) \,:\, k \in \mathcal{K},\, i \in \mathcal{I}\big) \quad (2)$$

We denote with $\mathcal{N}_k \subseteq \mathcal{N}$ the subset of BSs that are in range with user class k and assume that user association can be accomplished based on network performance or cost criteria.

B. Problem Statement

Operator Objective. The objective of the network operator is to deliver as many of the requested video files as possible, with the minimum delay and the minimum possible servicing cost. These latter depend on the demand and the servicing policy of the operator, i.e., the base stations and the backhaul links that will be used. We denote with $J_n(a) \geq 0$ the cost incurred by the MNO when it uses SCBS $n \in \mathcal{N}$ to deliver content with rate of $a \geq 0$ bps, and $J_M(a)$ the respective cost for the MBS. This cost includes the BS energy consumption and is positively correlated to the distance between the BS and the served users.

Since the resources of the operator are limited, some user requests may not be served or served with practically intolerable delays. This induces cost to the operator due to future revenue losses, e.g., unsatisfied clients unsubscribe from the service. We introduce a penalty function $P(\cdot)$ to capture this cost, which is assumed to be a positive, increasing and convex function of the number of unserved requests.

User Experienced Delay. The average delay D_{ki} experienced by user class k for downloading item $i \in \mathcal{I}$ depends on the path and the congestion of the respective links. The main components of the delay are the processing, propagation and transmission delay as well as the queueing delay which captures link congestion [18]. For each user class k the operator determines the portion of the requests that will be routed over each possible path leading either to a cache of an SCBS (having the item), or to a content server. Clearly, the routing decisions of the operator are coupled with its caching policy.

The MNO decides whether it will cache a certain file and with what quality at each SCBS. Different versions (or layers) of each file have different size and can satisfy different subset of requests. In order to serve a user by a SCBS, the requested content should either be already cached there or fetched via the respective backhaul link. This latter option adds delay which, depending on the backhaul type, capacity and load, may be quite significant[7]. Alternatively, the MBS can serve the requests with smaller delay (if it is not heavily loaded).

The delay minimization and the cost minimization may in general be conflicting objectives, and hence need to be balanced properly, depending on the objective of each operator. Formally, the problem of the MNO can be defined as follows.

MNO Video Delivery Problem (MVD). *Given: (1) the matrix* $\mathbf{\Lambda}$ *of requests for video files, for a certain time period, (2) the storage and average capacities of the SCBSs,* S_n, C_n, G_n, $\forall n \in \mathcal{N}$, *(3) the servicing cost of the BSs,* $J_M(\cdot)$, $J_n(\cdot)$, $\forall n \in \mathcal{N}$, *and (4) the penalty cost* $P(\cdot)$ *for rejecting requests:*

- *for each video file, decide in which base stations it will be cached and at which quality (which version/layers),*
- *for each request from user class* k*, determine from which base stations it will be served, whether backhaul links will be used or if it will be delivered by the MBS, and with what quality,*

so as to optimize the balanced objective of average user experienced delay, and total servicing and penalty cost.

[4]The ordering is wrt quality: with slight abuse of notation, we use the index of the layer (and the version) to denote also the respective quality.

[5]See for example, Vidyo: http://www.vidyo.com, and Radvision: http://www.radvision.com.

[6]Notice that this is not a restriction or an assumption, rather it implies that we group the users based on location and on quality requirements.

[7]According to estimations from industry [2], if the backhaul link adds delay less than 1ms, then it can be considered negligible.

III. Delivering Versions

In this section we formally introduce the problem for the case of video encoding in multiple qualities (versions) and accordingly present a methodology for its solution.

A. MVD Problem Formulation

Decision Variables and Constraints. Let $x_{niv} \in \{0,1\}$ denote whether version $v \in \mathcal{V}$ of file $i \in \mathcal{I}$ will be cached at SCBS $n \in \mathcal{N}$. These variables consitute the caching policy of the MNO:

$$\boldsymbol{x} = \left(x_{niv} \,:\, n \in \mathcal{N},\, i \in \mathcal{I},\, v \in \mathcal{V}\right) \tag{3}$$

Also, the variable $y_{kniv} \in [0,1]$ denotes the portion of requests of user-class k for file i that will be satisfied with version v downloaded from SCBS $n \in \mathcal{N}$, and $y_{kMiv} \in [0,1]$ is the respective decision for the MBS (M). Finally, $z_{kniv} \in [0,1]$ is the portion of requests of user k for file i that require fetching version v via the backhaul of SCBS $n \in \mathcal{N}$. The routing policy of the MNO is described by the following matrices:

$$\boldsymbol{y} = \left(y_{kniv},\, y_{kMiv} \,:\, k \in \mathcal{K},\; n \in \mathcal{N}_k,\; i \in \mathcal{I},\; v \in \mathcal{V}\right) \tag{4}$$

$$\boldsymbol{z} = \left(z_{kniv} \,:\, k \in \mathcal{K},\, n \in \mathcal{N}_k,\, i \in \mathcal{I},\, v \in \mathcal{V}\right) \tag{5}$$

In order for a SCBS to send a version to a user, it needs either to have it cached or to download it through the backhaul. Moreover, the servicing rate from SCBS n to user k can be maximum if item i is cached, and if not, it cannot exceed backhaul servicing rate. Hence:

$$y_{kniv} \leq x_{niv} + z_{kniv} \;\; \forall\, k \in \mathcal{K},\, n \in \mathcal{N}_k,\, i \in \mathcal{I},\, v \in \mathcal{V} \tag{6}$$

Besides, the MNO cannot satisfy a request more than once:

$$\sum_{v \geq q_{ki}} y_{kMiv} + \sum_{n \in \mathcal{N}_k} \sum_{v \geq q_{ki}} y_{kniv} \leq 1, \;\; \forall\, k \in \mathcal{K},\, i \in \mathcal{I} \tag{7}$$

Notice that the requests of each user class for each file may be satisfied by versions of different qualities (higher or equal to the minimum quality required by the user). Hence, there is a need to add the respective components.

Clearly, the routing and caching policies are constrained by the wireless capacity and storage capacity of the base stations. Also, for the routing decisions there may be additional constraints due to interference. The current industry practice is to use a disjoint channel allocation across different tiers of base stations [13], as well as orthogonal subchannels for overlapping base stations within each tier [15]. Channel orthogonality across different BSs can also be provided with the introduction of almost blank subframes (ABS) that were defined in the eICIC mechanism. The impact of this type of interference management techniques on the system's (average) capacity can be modeled using the protocol interference model [19]. Clearly, the data delivery of interfering SCBSs cannot be concurrently maximized. Specifically, it should hold:

$$\sum_{n \in \mathcal{N}_k} \frac{1}{C_n} \sum_{m \in \mathcal{K}} \sum_{i \in \mathcal{I}} \sum_{v \in \mathcal{V}} y_{mniv} o_{iv} \lambda_{mi} \leq 1, \;\; \forall\, k \in \mathcal{K} \tag{8}$$

where $\mathcal{N}_k$ is the set of SCBS in range with user k and hence are potential interferers. Due to space limitations, we refer the reader to our online technical report [20] for further details.

Delay-Cost Minimization. The delay cost of user class k requesting file i, depends on caching $\boldsymbol{x}$ and routing policy $(\boldsymbol{y}, \boldsymbol{z})$:

$$D_{ki}(\boldsymbol{y}, \boldsymbol{z}, \Lambda) = \sum_{v \geq q_{ki}} \left(d_{kMiv}(\boldsymbol{y}, \Lambda) + \sum_{n \in \mathcal{N}_k} d_{kniv}(\boldsymbol{y}, \boldsymbol{z}, \Lambda) \right) \tag{9}$$

where d_{kniv} is the delay when users k are served by SCBS $n \in \mathcal{N}_k$ with version v of file i, and d_{kMiv} is the delay when served by the MBS. Notice that the summation over the different versions is necessary since a request can be satisfied with higher quality version, e.g., if it is already available in a nearby SCBS. The flow-level delay we consider here has fixed components, such as the propagation and the processing delay at the base stations (or routers), and some components that vary with the load of the links. The latter is often modeled as a queueing delay under the hypothesis of $M/M/1$ queueing behavior [18]. Namely, each bit traversing a link of capacity C bps which carries a total flow with rate $f \geq 0$ bps, experiences a delay of $1/(C-f)$ seconds.

The MNO servicing cost $J_n(\boldsymbol{y}, \boldsymbol{z}, \Lambda)$ depends on the total bandwidth that each SCBS n delivers to the subscribers, including the backhaul link consumption. Similarly, for the MBS, it is $J_M(\boldsymbol{y}, \Lambda)$. Notice that we do not take into account the RAN backhaul cost since this hop is common for the SCBS and the MBS. We assume that these functions are positive, increasing and convex on the delivered bandwidth due to congestion effects [8][8]. The aggregate servicing cost of the operator can be defined as follows:

$$J(\boldsymbol{y}, \boldsymbol{z}, \Lambda) = J_M(\boldsymbol{y}, \Lambda) + \sum_{n \in \mathcal{N}} J_n(\boldsymbol{y}, \boldsymbol{z}, \Lambda) \tag{10}$$

The goal of the MNO is to minimize, for a certain time period of duration T, the average delay for all users and the total servicing and penalty cost. This is achieved by solving the following joint routing and caching optimization problem (*MVD Problem*):

$$\min_{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}} J(\boldsymbol{y}, \boldsymbol{z}, \Lambda) + P(\boldsymbol{y}, \Lambda_P) + \alpha \sum_{k \in \mathcal{K}} \sum_{\in \mathcal{I}} D_{ki}(\boldsymbol{y}, \boldsymbol{z}, \Lambda)$$

s.t. (6), (7), (8)

$$\sum_{i \in \mathcal{I}} \sum_{v \in \mathcal{V}} o_{iv} x_{niv} \leq S_n, \;\; \forall\, n \in \mathcal{N} \tag{11}$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \sum_{v \geq q_{ki}} \lambda_{ki} o_{iv} y_{kniv} \leq C_n T, \;\; n \in \mathcal{N} \tag{12}$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \sum_{v \geq q_{ki}} \lambda_{ki} o_{iv} z_{kniv} \leq G_n T \;\forall\, n \in \mathcal{N} \tag{13}$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \sum_{v \geq q_{ki}} \lambda_{ki} o_{iv} y_{kMiv} \leq C_M T \tag{14}$$

$$z_{kniv}, y_{kniv}, \; y_{kMiv} \in [0,1], \; \forall k \in \mathcal{K}, n \in \mathcal{N}_k, i \in \mathcal{I}, v \in \mathcal{V} \tag{15}$$

$$x_{niv} \in \{0,1\} \;\; \forall\, n \in \mathcal{N},\, i \in \mathcal{I},\, v \in \mathcal{V} \tag{16}$$

where parameter $\alpha > 0$ (measured in monetary units over time units) is determined by the operator and is used to balance the cost-delay objectives. For example, an operator interested in reducing the delay even at the expense of higher servicing cost

[8]The exact form of the backhaul link cost function depends on various technical parameters (e.g., transmission technology) and economic parameters (leased or owned) [8].

may set a high value for α. Also, there is no MBS backhaul constraint (assume that RAN backhaul is sufficiently large) and it is $y_{kniv} = 0$ when $n \notin \mathcal{N}_k$. Finally, Λ_P is the number of unserved requests:

$$\Lambda_P = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} \lambda_{ki}\big(1 - \sum_{v\geq q_{ki}} [y_{kMiv} + \sum_{n\in\mathcal{N}_k} y_{kniv}]\big)$$

The objective function of the MVD problem is convex under the assumptions about the properties of $D_{ki}(\cdot)$, $J(\cdot)$ and $P(\cdot)$. However, the constraint set includes the $0-1$ decisions variables x_{niv} that render it *NP-hard* (for the proof please see our online technical report [20]). In the sequel, we present an algorithm for deriving a solution that asymptotically converges to the optimal one.

B. MVD Solution Method

In order to solve the MVD problem we use the method of Lagrange partial relaxation [21]. Specifically, we relax the constraints in (6) and introduce the respective set of dual Lagrange multipliers:

$$\boldsymbol{\mu} = (\mu_{kniv} \geq 0 : \forall\, k \in \mathcal{K},\, n \in \mathcal{N}_k, i \in \mathcal{I}, v \in \mathcal{V}) \quad (17)$$

This relaxation simplifies the solution of the problem and admits an intuitive interpretation since it decouples the routing and the caching decisions of the operator.

First, we define the Lagrange function as follows:

$$L = J(\boldsymbol{y}, \boldsymbol{z}, \Lambda) + P(\boldsymbol{y}, \Lambda_P) + \alpha \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} D_{ki}(\boldsymbol{y}, \boldsymbol{z}, \Lambda) + \\ + \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}_k} \sum_{i\in\mathcal{I}} \sum_{v\in\mathcal{V}} \mu_{kniv}(y_{kniv} - x_{niv} - z_{kniv})$$

Using this relaxation, the problem can be rewritten:

$$\max_{\boldsymbol{\mu}} \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \quad L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\mu}) \\ s.t. \quad (7), (8), (11), (12), (13), (14), (15), (16) \\ \mu_{kniv} \geq 0,\, \forall k \in \mathcal{K}, n \in \mathcal{N}_k,\, i \in \mathcal{I}, v \in \mathcal{V} \quad (18)$$

which can be solved in an iterative fashion, using a primal-dual Lagrange method. Notice that due to the discrete constraint set, we have to employ a subgradient method for updating the dual variables [21].

In each iteration, denoted with t, the dual objective is improved using a subgradient update and accordingly the primal relaxed problem is solved in order to update the primal variables (which in turn are used in the subsequent dual objective update). The dual variables are updated with a subgradient method as follows:

$$\mu_{kniv}^{(t+1)} = [\mu_{kniv}^{(t)} + \sigma^{(t)} g_{kniv}^{(t)}]^+,\, \forall\, k \in \mathcal{K}, n \in \mathcal{N}, i \in \mathcal{I}, v \in \mathcal{V} \quad (19)$$

where $[.]^+$ denotes the projection onto the non-negative orthant, and $\sigma^{(t)}$ is the step size at iteration t. Also, $g_{kniv}^{(t)}$ is the subgradient, i.e., $g_{kniv}^{(t)} = y_{kniv}^{(t)} - x_{niv}^{(t)} - z_{kniv}^{(t)}$.

Then, we need to solve the relaxed primal problem and obtain the updated values of $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ (for the current iteration t). Interestingly, the primal problem can be further decomposed into two subproblems, named P_1 and P_2, as follows:

$$P_1 \quad : \quad \max_{\boldsymbol{x}} \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}_k} \sum_{i\in\mathcal{I}} \sum_{v\in\mathcal{V}} \mu_{kniv} x_{niv} \\ s.t. \quad (11), (16)$$

Algorithm 1: Primal-Dual Algorithm

Input : $J(\cdot)$, $P(\cdot)$, $D_{ik}(\cdot)$, α, $\boldsymbol{\Lambda}$
Output: $\boldsymbol{x}^*$, $\boldsymbol{y}^*$, $\boldsymbol{z}^*$

Initialize dual variables $\boldsymbol{\mu^1}$ to zero, the lower bound as $LB = -\infty$ and the upper bound as $UB = +\infty$.
$t \leftarrow 1$;
repeat
  Solve P_1 and find $\boldsymbol{x}^{(t)}$;
  Solve P_2 and find $\boldsymbol{y}^{(t)}$ and $\boldsymbol{z}^{(t)}$;
  Name as $q(\mu^{(t)})$ the solution value of the primal problem;
  if $q(\mu^{(t)}) > LB$ **then**
    $LB = q(\mu^{(t)})$;
  end
  Update UB;
  Update dual variables $\boldsymbol{\mu}^{(t+1)}$ using (19);
  $t \leftarrow t + 1$;
until $\frac{UB-LB}{UB} < 0.1$ *and* $t < 1000$;

and

$$P_2 \quad : \quad \min_{\boldsymbol{y},\boldsymbol{z}} J(\boldsymbol{y}, \boldsymbol{z}, \Lambda) + P(\boldsymbol{y}, \Lambda_P) + \\ + \alpha \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} D_{ki}(\boldsymbol{y}, \boldsymbol{z}, \Lambda) + \\ + \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}_k} \sum_{v\in\mathcal{V}} \mu_{kniv}(y_{kniv} - z_{kniv}) \\ s.t. \quad (7), (8), (12), (13), (14), (15)$$

P_1 involves only the caching variables $\boldsymbol{x}$. Hence, we call it the *caching subproblem*. Also, P_1 is separable into $|\mathcal{N}|$ unidimensional knapsack problems, one for each SCBS $n \in \mathcal{N}$, and thus can be solved in a distributed manner. The knapsack problem can be optimally solved in pseudo-polynomial time using dynamic programming methods. On the other hand, P_2 involves only the bandwidth allocation decisions $\boldsymbol{y}$ and $\boldsymbol{z}$. We call it the *bandwidth allocation* subproblem. The objective function of P_2 is strictly convex and the constraint set is convex, compact and continuous. Hence, it can be efficiently solved using standard convex optimization techniques [21]. For more details about the solution method please refer to the online technical report [20]. The method is summarized in Algorithm 1.

The solution that we provide for this NP-hard problem converges asymptotically to the optimal solution. Specifically the following lemma holds:

Lemma 1. *Algorithm 1 converges asymptotically to the optimal solution* $\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*$.

Proof. The convergence of this type of primal-dual algorithms with subgradient updates (non-differentiable dual functions) is ensured if (i) a proper diminishing step size is selected satisfying conditions of Prop. 8.2.6 [21, Chapter 8], (ii) the subgradients are bounded. Here, we follow the methodology in [22] and set $\sigma^{(t)} = \nu \frac{UB - q(\mu^{(t)})}{||g^{(t)}||^2}$, where ν is a parameter with positive value and UB is the upper bound on each iteration that can be calculated by simply finding a feasible solution to

the primal problem. For more details about the calculation of the UB please refer to the online technical report [20]. Also, by their definition, it can be directly seen that the subgradients are bounded. □

An operator can use Algorithm 1 to solve offline, that is at the beginning of each time period, the MVD problem and find a near optimal joint routing and caching policy for versions. In the sequel, we extend our methodology for the case of layered encoding.

IV. Delivering Layers and Video Streaming

In this section, we explain how the system architecture and the video delivery optimization problem described previously change, when the MNO employs SVC for compressing and delivering the video files. Next, we discuss how the proposed optimization approach is related to the key video streaming parameters that are configured at the video decoder.

A. Layered Encoding

We extend the model to include video encoded in layers, and more specifically we adopt the SVC extension of the H.264/MPEG4-AVC standard [16]. The operator can deliver either a version or a subset of layers that satisfies the minimum quality requirement of user requests. One of our goals is to investigate under what conditions caching and delivering layers is less costly for the operator than delivering versions of the files. This is of major importance since layers are rarely used today by CDNs (and MNOs) due to the additional rate overhead they introduce. However, as we will explain in the sequel and demonstrate in the performance evaluation section, for HCNs with storage capable SCBSs, caching layers may be beneficial in terms of servicing cost and experienced delay instead of transcoding the video into multiple versions. The reason is that more flexible caching and routing decisions can be made, since each user can receive different layers (for the same file) through different paths. For example, the base layer can be delivered even via costly links if this ensures small delays, while subsequent layers (that improve quality) can be delivered through other, less costly and/or congested paths.

Before we formally describe the extension of our optimization framework, we have to clarify that SVC allows different types of scalable encoding (spatial, temporal, quality) to be combined and create a single layer. Our modeling approach for layered video is generic and it can capture the delivery of all the potential scalability combinations that are available in SVC. This is possible because every scalability combination can be expressed in terms of specific well-ordered dependencies between the involved layers. In this paper we are not concerned with the specifics of the layered encoding and the optimal selection of scalability combinations but only with the implications on network delivery of this relatively new type of encoded video streams.

The set of layers is denoted as $\mathcal{L} \triangleq \{1, 2, \ldots, Q\}$. These layers introduce additional constraints on the derivation of the MNO's policies. Namely, there is no benefit for a user to receive a specific layer if it has not received all the lower layers. Hence, for each user k requesting file i should hold:

$$\sum_{n \in \mathcal{N}_k} y_{kni(l+1)} = \sum_{n \in \mathcal{N}_k} y_{knil}, \ \forall\, l < q_{ki} \quad (20)$$

$$\sum_{n \in \mathcal{N}_k} y_{knil} = 0, \forall\, l > q_{ki} \quad (21)$$

where we extended the definition of $\boldsymbol{y}$ variables, so that they also refer to layers, i.e. $y_{knil} \in [0, 1]$ denotes the portion of requests of user-class k for file i that will be satisfied with layer l downloaded from BS $n \in \mathcal{N}$. We do the same for the rest of the optimization variables $\boldsymbol{x}$ and $\boldsymbol{z}$.

Requests can be satisfied either using layers or versions. In any case, each request by user k for each file i cannot be satisfied more than one time:

$$y_{kMil_q} + \sum_{v \geq q_{ki}} y_{kMiv} + \sum_{n \in \mathcal{N}_k} y_{knil_q} + \sum_{n \in \mathcal{N}_k} \sum_{v \geq q_{ki}} y_{kniv} \leq 1 \quad (22)$$

where $l_q \in \mathcal{L}$ is the highest layer required to satisfy quality request q_{ki}.

The delay components are similarly defined as before, with the only difference that there is a need to minimize the *maximum delay* for each delivered layer (since all layers must be delivered in order to watch the video). Namely, the delay D_{ki} is now written as follows:

$$D_{ki}(\boldsymbol{y}, \boldsymbol{z}, \Lambda) = \sum_{v \geq q_{ki}} (d_{Mikv}(\boldsymbol{y}, \Lambda) + \sum_{n \in \mathcal{N}_k} d_{nikv}(\boldsymbol{y}, \boldsymbol{z}, \Lambda)) + d_{kMil_d}(\boldsymbol{y}, \Lambda) + \sum_{n \in \mathcal{N}_k} d_{knil_d}(\boldsymbol{y}, \boldsymbol{z}, \Lambda) \quad (23)$$

where l_d is the layer that is delivered with the largest delay (for each user k and each file i). Notice also that some requests of each user for each file may be satisfied by layers while others by versions. Hence, there is a need to add the respective delay components.

B. Video Streaming Concerns

From the perspective of the end user the result of our optimization is the minimized delivery delay of the requested video file. This delay will directly impact the video streaming process through the necessary *startup delay* that is introduced at the video decoder of the user. In the sequel, we will describe how the benefits that our optimization framework provides can be readily translated into a minimized startup delay.

In typical video decoders there are two delay components that are introduced before the video playback commences. First, for playback to start there is a need to buffer a certain number of video frames that can be translated to either a portion of the file in bytes or seconds. Let us denote this inelastic buffering delay as D_{b_1}. Second, the video decoder usually adds an additional delay element before it starts the playback in order to accommodate fluctuations in the bandwidth of the network. This is usually exercised by measuring the RTT and the average receiver data rate. This additional delay component is denoted as D_{b_2}. Finally, when the video decoder commences the playback of the video, it will normally maintain a constant playback rate in terms of frames-per-second (fps). Thus, this delay component that accounts for the rendering of the video

file is denoted as D_r and is simply equal to the length of the video in seconds. What all the above mean is that once the user requests the video file, say at time instant $t = 0$, the time instant that the playback ends is equal to $D_{b_1} + D_{b_2} + D_r$.

From the last three delay components we discussed the only one that can be practically controlled is D_{b_2}. If at the decoder this delay is too small then the decoder might experience a *buffer underrun* which means that it requires data for decoding and playback but they have not yet been received. This is typically addressed with the undesired playback pauses. To avoid these events, the average delay that the complete video is delivered, must be lower than the time instant that the playback ends at the user decoder. Thus, the following condition must hold:

$$D_{ki}(\boldsymbol{y}, \boldsymbol{z}, \Lambda) \leq D_{b_1} + D_{b_2} + D_p \tag{24}$$

Thus, by minimizing the file delivery delay $D_{ki}(\boldsymbol{y}, \boldsymbol{z}, \Lambda)$, we can indirectly allow the video decoder to use a smaller delay D_{b_2}. For the defined system model, our optimization framework ensures the minimum startup delay in order to avoid a buffer underrun (for a specific cost-delay tradeoff expressed with parameter α).

Also, it is clear that the video streaming performance depends on the quality level of the delivered video, as this in turn determines the video file size. Hence, delivering versions of high quality is more likely to induce buffer underrun or other similar undesirable phenomena. On the other hand, layered encoding has a unique advantage as one can determine the video quality more dynamically. For example, deliver first the basic layer (with small size), and then, if the network conditions change, deliver the higher quality layer. This is particularly important for the case that the routing and caching decisions are taken not for the entire video files but for the video segments.

V. Performance Evaluation

In this section we present the numerical experiments that we have conducted to evaluate the performance of the proposed algorithm using realistic system settings. Our main objectives/goals are as follows:

- Compare the two different video encoding techniques, i.e. layered encoding and versions.
- Describe the cost-delay trade off.
- Examine the convergence of the proposed algorithm.

Methodology and Performance Criteria. Particularly, we compare the performance of our algorithm in three cases:

1) *Versions*: The files are encoded with different rates that yield multiple versions.
2) *Layered-Encoding*: Each file has a multi-layer representation based on SVC.
3) *Mixed Strategy*: The system supports both versions and layers.

The performance criteria that we consider are the incurred cost by the operator and the experienced delay by the users. Because of the quadratic relation between power consumption and achievable rate, we adopt the following cost function:

$$J_n(y, z, \Lambda) = \big(\sum_{k \in \mathcal{K}} w_{kn} \sum_{i \in \mathcal{I}} \sum_{q \in \mathcal{V} \cup \mathcal{L}} o_{iq} \lambda_{ki} y_{kniq} \big)^2 + \\ + \beta \big(\sum_{k=1}^{K} \sum_{i=1}^{I} \sum_{q \in \mathcal{V} \cup \mathcal{L}} o_{iq} \lambda_{ki} z_{kniq} \big)^2$$

where β is a positive constant and $w_{kn} \geq 0$ captures the wireless transmission efficiency among user k and base station n. The largest the value of w_{kn} is the less are the resources the network has to consume (e.g., frequency - time slots, or energy) in order to serve the user[9]. The experienced delay for each user k receiving version (or layer) q of file i by BS n is [6]:

$$d_{kniq}(\boldsymbol{y}, \boldsymbol{z}, \Lambda) = \frac{y_{kniq} \lambda_{ki} o_{iq}}{C_n - A_n(\boldsymbol{y})} + d_n^k y_{kniq} \lambda_{ki} o_{kniq} \\ + \frac{z_{kniq} \lambda_{ki} o_{iq}}{G_n - B_n(\boldsymbol{z})} + d_n^{bh} z_{kniq} \lambda_{ki} o_{kniq}$$

where $A_n(\boldsymbol{y}) = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \sum_{q \in \mathcal{V} \cup \mathcal{L}} (y_{kniq} \lambda_{ki} o_{iq})$ is the assigned load to the BS n. Similarly, for the backhaul link it is $B_n(\boldsymbol{z})$. d_n^k is the propagation and processing delay for the transmission from base station n to user k, and d_n^{bh} the respective delay component for the backhaul link of the base station.

We adopt the following linear penalty cost function for the unserved requests: $P(y, \Lambda_P) = \gamma \Lambda_P$, where Λ_P is the number of unserved requests, as defined in Section III, and γ is the unit cost incurred per unserved request.

Simulation Setup. Throughout, we consider a cellular network consisted of a single main base station located at the center of a circular-shaped cell with radius $200m$. $K = 200$ mobile users, $|N_{\mathcal{P}}| = 4$ $PBSs$ and $|N_{\mathcal{F}}| = 16$ $FBSs$ are uniformly placed in random statistically independent positions in the cell. The transmission radius of a PBS and a FBS is equal to $80m$ and $50m$ respectively. Neighboring BSs operate in orthogonal frequency bands. The coefficients w_{kn} are set as the fraction of the distance between the user k and the base station n over the radius of the cell. We also set $d_n^k = 1$ and $d_n^{bh} = 1$ for each base station n, $\beta = 1$ and $\gamma = 1000$.

In all simulations, we assume a collection of $M = 100$ files, each one of which can be delivered in $Q = 2$ quality levels. The size of a version of a file is equal to 10 and 20 units of data in the low and the high quality level respectively. The layered-encoding overhead is 10%. This is a realistic choice inspired by the video traces in [24]. Within a period T of size normalized to 1, each user requests a file based on a Zipf-Mandelbrot model [25] with a shape parameter value equal to 0.8 and a shift parameter value equal to 10. Unless otherwise specified, $C_n = 100$, $\forall n \in \mathcal{N} \cup M$, $S_n = 50$, $G_n = 100$, $\forall n \in \mathcal{N}$, $\alpha = 1$ and the requested video quality follows the uniform probability distribution. The parameter ν of Algorithm 1 is initially set to 2.0 and is halved if there is no improvement in the UB for 50 successive iterations [22].

[9]More formally, we can define the transmission efficiency as the average volume of traffic (measured in bits) that can be supported by one unit of spectrum resource (measured in Hz). Obviously, the transmission efficiency is closely related to the path loss and shadow fading of a link. In the simplest case, it can be defined as the Shannon capacity.

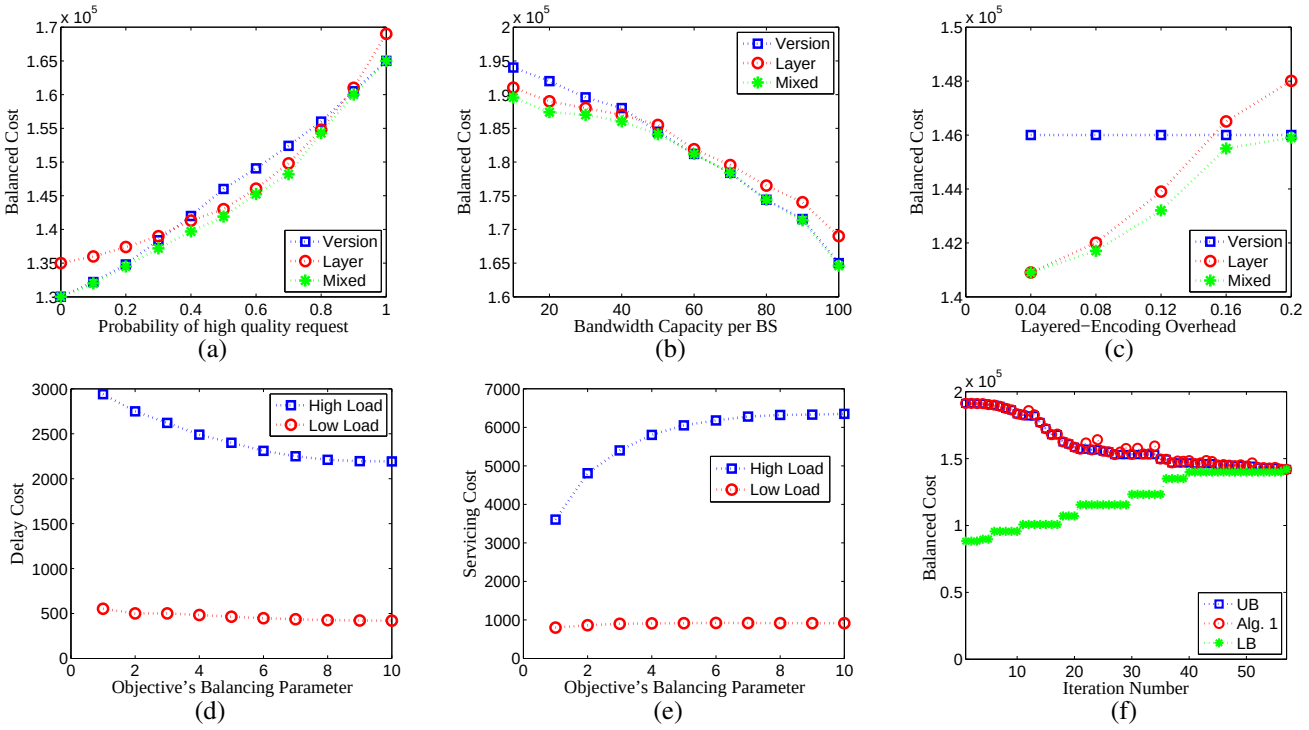

Fig. 2. The balanced cost of the operator as a function of (a) the probability that a request is for the high quality level, (b) the bandwidth capacity per BS and (c) the layered encoding overhead. The impact of the objective's balancing parameter α on (d) the delay cost and (e) the servicing cost. (f) The balanced cost of the operator, the UB and the LB at each step of the execution of Algorithm 1.

Comparison between Versions and Layers. We first compare the balanced cost of the operator achieved by the proposed algorithm as a function of the probability that a user request is for the high quality level in figure 2 (a). As expected, this cost increases when the aforementioned probability increases, as more users request the high quality level of the files and, thus, a larger amount of data is downloaded. We observe that *pure version caching is desirable when the requests are homogeneous in terms of the requested quality level*. This is because of the overhead that layered encoding incurs. However, when both the quality levels are requested layered encoding may be preferable, as it requires less storage space for serving the same requests compared to the pure versions scheme. Mixed strategy operates the best, as it efficiently combines the inherent features of the two encoding methods (e.g. the user demand is homogeneous in terms of the requested quality in a cell's subregion, but it is heterogeneous in the rest of it).

We then analyze the impact of the transmission bandwidth capacity of each SCBS on the balanced cost of the operator. Figure 2 (b) shows the results when all the requests are for the high quality level. We observe that for low capacity values the layers achieve lower cost than the pure version scheme. This is because *using layers balances the traffic across the base stations*, as the same user can fetch the two layers from two different neighboring base stations. This cost decrement is more crucial when the base station capacity is low. For higher values of the BS capacity, the layered encoding overhead can not be justified by the above gain and, thus, the pure versions caching scheme outperforms the layered scheme.

The superiority of the layered scheme compared with the versions scheme depends on the encoding overhead, as depicted in figure 2 (c). When this overhead becomes large enough, the pure version caching scheme becomes more preferable.

Study of the cost-delay trade off. The objective's balancing parameter α reflects the preference of the operator for reducing the user experienced delay or the servicing cost. Figures 2 (d)-(e) show the results in the versions case when the per user demand is 1 (Low Load) and 10 (High Load). *As α increases the user delay decreases, at the expense of the servicing cost increase and vice versa*. This is because when α is low, users are assigned to the SCBSs with the largest spectral efficiency, but when α increases they are forced to be assigned to the less loaded SCBSs regardless of the spectral efficiency (and thus consuming more energy). The impact of the parameter α on the two metrics is greater in the High Load case, as the assignment of the users to the overloaded base stations is more costly to the operator. We observe that improving the delivery delay (by tuning properly the parameter α) by an average 10% may increase the servicing cost from 10% up to 30%.

Convergence of Algorithm 1. The MVD problem is an NP-hard problem and hence its solution cannot be derived in polynomial time. The iterative solution we propose gradually improves the obtained result. In other words, a network operator can execute the suggested algorithm in an offline fashion to determine for each time period the joint routing and caching policy. The performance improves with the time that the algorithm runs. Specifically, the convergence of our algorithm in the versions case is depicted in Fig. 2 (f). Even when layers come into the picture, *the convergence typically happens in less than a few hundreds steps*.

VI. Related Work

The idea of leveraging in-network storage for improving network performance is gaining increasing interest[10] and has been recently proposed also for wireless networks [3]. We generalize this architecture for multiple tiers of base stations. Additionally, in contrast to [3], we consider the realistic case that the base stations have *congestible* (capacitated) links. This calls for joint derivation of routing and caching policies.

Similar policies have been studied before for wireline networks. For example, in [22] the authors studied the joint caching and request routing problem in content delivery networks (CDNs), and a similar study for IPTV networks was provided in [9]. Also, the authors of [10] proposed a joint content placement and routing algorithm that leverages small CDN servers installed within users' homes. Unlike these studies, the proposed model here allows each content item to be cached (i) with different quality, and (ii) at BSs in different tiers. Additionally, in HCNs the BSs at the different tiers have overlapping coverage areas and are not connected each other.

Network-aware video delivery mechanisms have been long studied (e.g. see [23] and references therein). In our setting, the network architecture is multi-tier, hence there are multiple paths that can be used to satisfy the requests of the users. Also, the bottleneck is often at the backhaul links and not on the last-hop wireless links as it was the case in conventional cellular networks. The cost and the impact of backhaul links on overall performance (delay) is very important [2] and has not been studied yet. We explicitly take into account this aspect. Finally, in HCNs, the different type of base stations incurs different cost for the operator (e.g., due to different energy consumption), and this has to be taken into account in designing the routing and caching policy.

Recently, video delivery over HCNs has been considered [26], [27]. Our model differs from these works since they do not take caching decisions, do not consider delivering video files from base stations in different tiers, nor the scenario of fetching requested items on demand through the (capacitated) backhaul links of the small cells. Besides, we consider hard quality constraints for user requests, and we investigate the performance - cost tradeoff for the MNO.

VII. Conclusions

We studied optimal joint routing and caching policies for multiple-quality video delivery in heterogeneous cellular networks. This is a problem of increasing importance as currently the explosion of mobile video traffic challenges the cellular operators. Our work reveals the potential benefits of incorporating two basic encoding techniques into the video delivery process. Moreover, it explores the cost-delay trade off for different system parameters and users' demand, that enables the operators to determine the optimal balance based on their priorities. The presented framework is generic in the sense that it allows to investigate the performance impact of various network architectures. As a topic of future work, we plan to evaluate our methods in the wireless testbed Nitos [28].

[10]See "A survey of in-network storage systems", IETF, http://tools.ietf.org/html/rfc6392.

ACKNOWLEDGEMENT

This work was funded by the research program 'CROWN', through the Operational Program 'Education and Lifelong Learning 2007-2013' of NSRF, which has been co-financed by EU and Greek national funds.

References

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update", Feb. 2012.

[2] Small Cell Forum, "Backhaul Technologies for Small Cells: Use Cases, Requirements and Solutions", *Technical Report/White Paper*, Feb. 2013.

[3] N. Golrezaei, et al, "FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers", *in Proc. of IEEE Infocom*, 2012.

[4] M. Zinka, K. Suhb, Y. Gua, and J. Kurose, "Characteristics of YouTube Network Traffic at a Campus Network - Measurements, Models, and Implications", *Computer Networks*, vol. 53, no. 4, 2009.

[5] Intel, "Rethinking the Small Cell Business Model", White Paper, 2011.

[6] K. Son, H. Kim, Y. Yi, and B. Krishnamachari, "Base Station Operation and User Association Mechanisms for Energy-Delay Tradeoffs in Green Cellular Networks", *IEEE JSAC*, vol. 29, no. 8, 2011.

[7] O.Arnold, F.Richter, G.Fettweis, and O.Blume, "Power Consumption Modeling of Different Base Station Types in Heterogeneous Cellular Networks", *in Proc. ICT MobileSummit*, 2010.

[8] W. Jiang, R. Zhang-Shen, J. Rexford and M. Chiang, "Cooperative content distribution and traffic engineering in an ISP network", *in Proc. of ACM SIGMETRICS*, 2009

[9] J. Dai, Z. Hu, B. Li, J. Liu, and B. Li, "Collaborative Hierarchical Caching with Dynamic Request Routing for Massive Content Distribution", *in Proc. of IEEE Infocom*, 2012.

[10] J. W. Jiang, S. Ioannidis, L. Massoulie, and F. Picconi, "Orchestrating Massively Distributed CDNs", *in Prof. of ACM CoNEXT*, 2012.

[11] X. Bao, et al, "DataSpotting: Exploiting Naturally Clustered Mobile Devices to Offload Cellular Traffic", *IEEE Infocom*, 2013.

[12] A. Ghosh, *et al*, "Heterogeneous Cellular Networks: From Theory to Practice", *IEEE Comm. Magaz.*, vol. 50, no. 6, 2012.

[13] Y. Kishiyama, A. Benjebbour, H. Ishii, and T. Nakamura, "Evolution Concept and Candidate Technologies for Future Steps of LTE-A", *in Proc. of IEEE ICCS*, 2012.

[14] W. Cheung, T. Quek, and M. Kountouris, "Throughput Optimization, Spectrum Allocation, and Access Control in Two-Tier Femtocell Networks", *IEEE JSAC*, vol. 30, no. 3, 2012.

[15] D. Astely, et al., "LTE: The Evolution of Mobile Broadband", *IEEE Communications Magazine*, vol. 47, no. 4, 2009.

[16] H. Schwartz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 17, no. 9, 2007.

[17] F. Hartanto, J. Kangasharju, M. Reisslein, K. W. Ross, "Caching video objects: layers vs versions?", *Multimedia Tools Appl.*, vol. 31, no. 2, 2006.

[18] D. P. Bertsekas, R. Gallager, "Data Networks", Athena Scientific, 2003.

[19] A. Karnik, A. Iyer, C. Rosenberg, "What is the right model for wireless channel interference?", *IEEE Trans. on Wirel. Comm.*, vol. 8, no. 5, 2009.

[20] K. Poularakis et al, "Video Delivery over Heterogeneous Cellular Networks: Optimizing Cost and Performance", *Technical Report, available at http://georgeiosifidis.net/wp-content/uploads/2011/03/PIAT-Video-Delivery-over-HCNs.pdf*.

[21] D. Bertsekas, A. Nedic, and A. Ozdaglar, "Convex Analysis and Optimization", *Athena Scientific Press*, 2003.

[22] T. Bektas, et al., "Exact Algorithms for the Joint Object Placement and Request Routing Problem in Content Distribution Networks", *Comp. and OR*, vol. 35, no. 12, 2008.

[23] J. Huang, Z. Li, Mung Chiang, A. Katsaggelos,, "Joint Source Adaptation and Resource Allocation for Multi-User Wireless Video Streaming", *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 18, no. 5, 2008.

[24] Video Trace Library: http://trace.eas.asu.edu/

[25] M. Hefeeda and O. Saleh, "Traffic Modeling and Proportional Partial Caching for Peer-to-Peer Systems", Trans. Netw., vol. 16, no. 6, 2008

[26] D. Bethanabhotla, G. Caire, and M. Neely, "Joint Transmission Scheduling and Congestion Control for Adaptive Streaming in Wireless Device-to-Device Networks", *in Proc. of IEEE Asilomar*, 2012.

[27] V. Joseph, and G. de Veciana, "Jointly Optimizing Multi-user Rate Adaptation for Video Transport over Wireless Systems: Mean-Fairness-Variability Tradeoffs", *in Proc. of IEEE Infocom*, 2012.

[28] NITlab, nitlab.inf.uth.gr.