Jedari, Behrouz; Di Francesco, Mario

Delay Analysis of Layered Video Caching in Crowdsourced Heterogeneous Wireless Networks

Published in:
2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings

DOI:
10.1109/GLOCOM.2018.8647205

Published: 20/02/2019

Document Version
Peer reviewed version

Please cite the original version:
Jedari, B., & Di Francesco, M. (2019). Delay Analysis of Layered Video Caching in Crowdsourced Heterogeneous Wireless Networks. In 2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings (pp. 1-6). [8647205] Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/GLOCOM.2018.8647205

Delay Analysis of Layered Video Caching in Crowdsourced Heterogeneous Wireless Networks

Behrouz Jedari and Mario Di Francesco
Department of Computer Science, Aalto University, Finland
Email:{behrouz.jedari,mario.di.francesco}@aalto.fi

***Abstract*—Caching popular content at small-cell base stations (SCBSs) and user equipments (UEs) can significantly reduce the network backhaul traffic while improving user satisfaction. This is also enabled by novel video encoding techniques, such as scalable video coding (SVC), which combine layers to offer content with different qualities without re-encoding. Despite some recent works, the performance of layered video delivery in crowdsourced heterogeneous networks (HetNets) is still unexplored. This article provides an analytical characterization the delay of video delivery in a network with multiple cache-enabled SCBSs and UEs, each storing part of the available video layers based on their popularity. Accordingly, video requests from an UE can be served by either SCBSs or UEs nearby. Our main objective is to maximize the cache hit probability by caching appropriate video layers, thereby minimizing the average video delivery delay. We formulate the problem of minimizing the delivery delay of layered video caching as an integer linear program. We then apply the difference of convex functions technique to identify the set of optimal video layers to be cached at each SCBS and UE in an iterative manner. Our results obtained by using a real video dataset demonstrate that our proposed solution significantly reduces the video download time of all UEs in the network.**

I. Introduction

The global mobile traffic is expected to grow from 7.2 exabytes per month in 2016 to 49 exabytes by 2021, where video data will account for 78% of the mobile traffic [5]. To cope with such a traffic growth, mobile network operators (MNOs) aim at expanding their network capacity by using different strategies, such as acquiring more radio spectrum or extending network infrastructure, both of which are often costly and time-consuming. Heterogeneous networks (HetNets) have emerged as a cost-effective alternative solution to increase the network capacity and alleviate the network backhaul traffic. In HetNets, cache-enabled small-cell base stations (SCBSs) with high-speed transmission technologies (e.g., LTE-A) are deployed at strategic locations to provide high-quality data delivery services and improve user satisfaction [11]. With the advent of mobile crowdsourcing, user-provided networking (UPN) also becomes part of HetNets where user equipments (UEs) in proximity serve content to each other through high-speed interfaces (e.g., Wi-Fi), eventually reducing network congestion and the cost of Internet access [17].

Several video caching strategies have been proposed in HetNets (see [11] for a survey). A number of studies leveraged the popularity of content to increase the cache hit rate probability. For instance, Golrezaei et al. [7] showed that caching popular videos in SCBSs can significantly improve the system throughput without deploying any additional infrastructure. Li et al. [12] evaluated multi-bitrate video caching, where a video is encoded in different qualities, each divided into multiple segments (or chunks). Once a user requests a video, the appropriate segments are dynamically fetched according to the current wireless link quality. Chen et al. [4] took user mobility into account and proposed a geographic caching strategy to maximize the cache hit probability at both SCBSs and UEs. The results showed that the reliability of data transmission between UEs considerably affects the caching throughput.

A limited number of recent studies have addressed caching layered videos, particularly, based on scalable video coding (SVC) [15]. SVC is an emerging video encoding technique over HTTP networks, which is an extension of the H.264/MPEG-4 encoding. SVC-based (or layered) video encoding is very suitable for wireless communications because it can adapt to fast varying wireless links without re-encoding [6]. The SVC format of a video includes a basic layer for a low-quality demand, which can be combined with several enhancement layers to provide users with high-quality videos. However, caching layered videos is complex since the dependencies between the different layers should be considered during both the caching and the streaming process. In this respect, Xie et al. [18] proposed an energy-efficient placement approach for layered videos with the aim of minimizing the energy saving of the SCBSs in cellular networks. Poularakis et al. [14] studied layered video caching in multi-operator cellular networks and found that cooperation of co-located SCBSs reduces the delivery delay up to 25%. However, prior work did not address minimizing the download delay of layered videos when SCBSs and UEs jointly perform content caching and delivery.

This article studies the problem of minimizing the delivery delay for layered video caching in crowdsourced HetNets. Each SCBS in the network caches a subset of the layers for each video based on its popularity, so as to maximize the number of UE requests that can be served through the cached video layers. In addition, UEs cache layered videos based on both popularity and their location to maximize the average hit rate probability of user requests served within the UPN. In summary, the major contributions of this work are the following.

- We provide an analytical model for the delay of layered video caching in crowdsourced HetNets based on the popularity of videos and the size of network caches.

- We formulate the problem of minimizing the delay of layered video caching as an integer linear programming problem, then apply the difference of convex (DC) functions technique to identify the video layers cached at each SCBS and UE.
- Numerical results demonstrate that our proposed solution significantly reduces the average download time of layered videos.

II. System Model

In this section, we model video caching in crowdsourced HetNets which can be supplied by an MNO. The key notation used is summarized by Table I.

A. Network and User Equipment

As shown in Fig. 1, we consider a wireless network including a macro base station (MBS) M connected to the core network through a high-capacity backhaul link (e.g., fiber optics). The transmission capacity and range of M are denoted as $c_M^w{\geq}0$ Mbps and $r_M{>}0$ meters, respectively. The network includes a set $\mathcal{S}{=}\{1,2,\dots,S\}$ of S small-cell base stations (SCBSs) and a set $\mathcal{U}{=}\{u_1,u_2,\dots,u_U\}$ of U user equipments (UEs) with caching capabilities. The MBS covers all SBSs and UEs in the network. The SCBSs are uniformly placed in $|S|$ locations that are connected to the core network through a wired or wireless backhaul link. The transmission capacity and range of each SCBS in $\mathcal{S}$ are denoted as $c_s^w{\geq}0$ Mbps and $r_s{>}0$ meters, respectively. The caching capacity of each SCBS s is indicated as $c_s{>}0$ MB. We assume that orthogonal frequency bands are assigned to SCBSs to manage the interference between them.

The location of users is described through a homogeneous Poisson point process (PPP) with intensity λ_U. We assume that at least one SCBS covers each $u_i{\in}\mathcal{U}$, where u_i only communicates with its nearest associated SCBS s. Two UEs communicate with each other over unlicensed bands (e.g., by using Wi-Fi Direct) when their distance is less than r_u (meters). The UEs can simultaneously access the Internet through cellular and Wi-Fi connections [16]. The average Internet download capacity of each $u_i{\in}\mathcal{U}$ through cellular connection c and Wi-Fi channel $f{\in}F$ are denoted as $c_{ui}^c{\geq}0$ and $c_{ui}^f{\geq}0$ Mbps, respectively. Moreover, the average transmission capacity of link $(u_i,u_j){\in}E^t$ through Wi-Fi channel $f{\in}F$ is denoted as $c_{ui,uj}^f{\geq}0$ Mbps. We assume that each UE can store different amount of data, where the caching capacity of $u_{i\in\mathcal{U}}$ is denoted as $c_{ui}{>}0$ MB.

The UEs create a mesh network that is modeled by a time-varying graph, where $G^t{=}(U,E^t,\xi^t)$ represents the graph at fixed-length time slot $t{\in}\mathcal{T},\mathcal{T}{=}1,2,\dots,T$. In G^t, $E^t{=}\{(u_i,u_j)|i,j{\in}U,\ i{\neq}j\}$ denotes the set of communication links and $\xi^t{=}\{(u_i,u_j)|i,j{\in}U,\ i{\neq}j\}$ denotes the set of links subject to interference between u_i and u_j at time t. In other words, the existence of $(u_i,u_j){\in}\xi^t$ implies that simultaneous data transmissions by users u_i and u_j at time t interfere with each other.

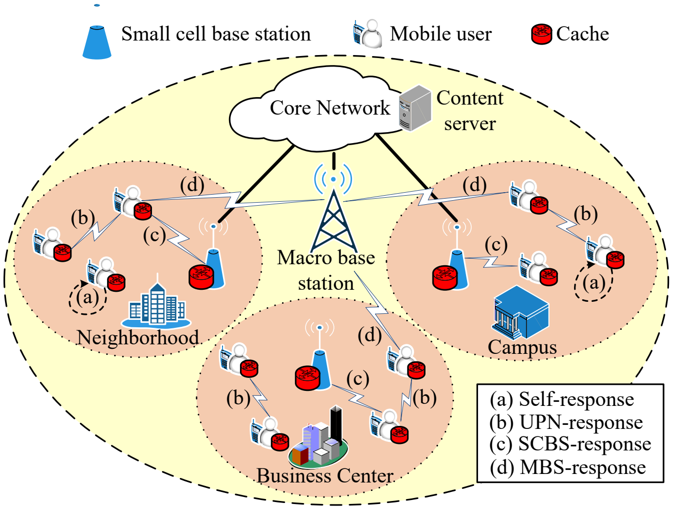


Fig. 1: The system model.

B. Video Encoding and User Demand

The MNO owns a finite set $\mathcal{V}{=}\{1,2,\dots,V\}$ of V video files, where each video is encoded with scalable video coding (SVC) [15]. Accordingly, each video consists of a set $\mathcal{Q}{=}\{1,2,\dots,Q\}$ of Q layers (or qualities) including one base layer and Q$-$1 enhancement layers; layer 1 realizes quality 1, the combination of layers 1 and 2 realizes quality 2, and so on. Unlike [18], we assume that the size of individual video layers can differ: the size of q-th layer of video v, denoted as $o_{vq}{>}0$ (MB), typically decreases with q $(o_{v1}{>}o_{v2}{>}\dots{>}o_{vQ})$. When a user requests the q-th quality level of video v, all layers of v from layer 1 up to layer q should be delivered to the user, where the size of the q-th quality level of v is $O_{vq}=\sum_{l=1}^{q}o_{vl}$.

Similar to some existing works (e.g., [7]), we identify the demand vector of each UE based on the file popularity, which can be characterized through a Zipf-like distribution. Such a popularity is available to the system, for instance, as predicted by employing learning methods [2]. We assume that the popularity of videos does not significantly change within a certain period (e.g., few hours or days), according to recent analysis of real video traces [13]. Thus, we sort the files in $\mathcal{V}$ in the descending order of their popularity and calculate the popularity of the i-th ranked video in $\mathcal{V}$ as:

$$p_i=\frac{i^{-\gamma}}{\sum_{j=1}^{V}j^{-\gamma}} \tag{1}$$

where γ is the skewness of the popularity distribution, which typically ranges between 0 and 1. For instance, $\gamma{=}0$ implies that all videos have the same popularity, where γ near to 1 implies that a few videos are viewed by a large number of UEs. We also denote each UE's video demand rate as ζ_u and the probability of requesting the q-th quality level of each video as $l_q{>}0$ $(q{=}1,2,\dots,Q)$.

C. Video Caching and Dissemination

Without loss of generality, we analyze layered video caching in a network with one SCBS s and its associated

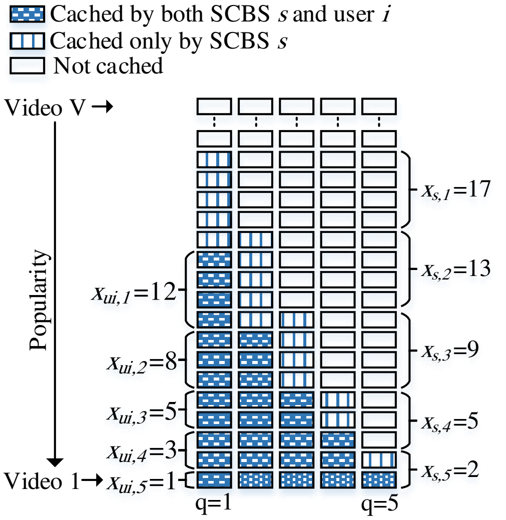


Fig. 2: A sample video placement in SCBS s and user u_i.

UEs in $\mathcal{N}\subseteq\mathcal{U}$. As a general principle, SCBS s or each $u_i\in\mathcal{N}$ cannot cache the q-th layer of video v unless all its previous layers (i.e., layer 1 up to layer $q-1$) have already been cached locally [14]. We define the placement for SCBS s as the set of parameters $X_s=x_{s,Q}\leq x_{s,Q-1}\leq\ldots\leq x_{s,1}\leq V$ $(x_{s,Q}\geq 0)$, where s caches Q layers of the $x_{s,Q}$ most popular videos, $Q-1$ layers of the videos with popularity between $x_{s,Q-1}$ and $x_{s,Q}$, and so on. Furthermore, the placement parameters should be related to the storage capacity of SCBS s as expressed by the following constraint:

$$\sum_{j=1}^{Q}\sum_{k=1}^{n(s,j)}\sum_{q=1}^{j} o_{z(s,j)+k,q}\leq c_s \tag{2}$$

The term $n(s,j)$ in Eq. (2) denotes the number of videos with the j-th quality level cached by SCBS s:

$$n(s,j)=\begin{cases} x_{s,j} & \text{if } j=Q\\ x_{s,j}-x_{s,j+1} & \text{if } 1\leq j<Q\end{cases} \tag{3}$$

while $z(s,j)$ is the index of the $(j+1)$-th placement parameter in X_s:

$$z(s,j)=\begin{cases} 0 & \text{if } j=Q\\ x_{s,j+1} & \text{if } 1\leq j<Q\end{cases} \tag{4}$$

Similar to SCBS s, we define the set of video placement parameters for $u_i\in\mathcal{N}$ as $X_U=[X_{u1},X_{u2},\ldots,X_{uN}]$. Here, the placement parameters of each u_i are defined as $X_{ui}=x_{ui,Q}\leq x_{ui,Q-1}\leq\ldots\leq x_{ui,1}\leq V$,$(x_{i,Q}\geq 0)$ and are subject to the following constraint:

$$\sum_{j=1}^{Q}\sum_{k=1}^{n(u_i,j)}\sum_{q=1}^{j} o_{z(u_i,j)+k,q}\leq c_{ui}, \forall i=1,\ldots,|\mathcal{N}| \tag{5}$$

where $n(u_i,j)$ and $z(u_i,j)$ are given similar to Eqs. (3) and (4), respectively.

TABLE I: Used symbols and their meaning

Symbol(s)	Definition
c^w_M, r_M	Transmission capacity and range of MBS M
$\mathcal{S}$, s	Set of SCBSs and a single SCBS
c^w_s, r_s	Transmission capacity and range of SCBS s
c_s	Storage capacity of SCBS s
$\mathcal{U}$, u_i	Set of UEs and a single UE u_i
λ_U	Density of UEs in PPP distribution
c^c_{ui}, c^f_{ui}	Cellular and Wi-Fi transmission capacity of u_i
c_{ui}, r_u	Storage capacity and transmission range of u_i
$c^f_{ui,uj}$	Wi-Fi transmission capacity between u_i and u_j
$\mathcal{V}$, v	Set of videos and a single video
Q	Number of video layers or qualities
o_{vq}	Size of the q-th layer of video v
ζ_u	User demand rate per time slot
l_q	Probability of requesting the q-th quality level of a video
p_i, γ	Popularity of the i-th ranked video and its skewness

Fig. 2 illustrates a possible placement of V videos – each with 5 layers – in SCBS s and a given UEs u_i associated to s. The figure shows that more layers of the most popular videos are cached because they are more frequently requested by the UEs. Specifically, SCBS s caches the 5-th quality level of video 1, the 4-th quality level of videos 2 and 3, the 3-rd quality level of videos 4 and 5, the 2-nd quality level of videos 6-8, and the 1-st quality level of videos 9-12. It is worth noting that the number of video layers cached by SCBS s is higher than those in u_i because the caching capacity of s is usually larger than that of UEs.

Once SCBS s and each $u_{i\in N}$ caches the video layers based on the placement policy, video offloading operates as follows. When u_i requests the q-th quality level of video $v\in\mathcal{V}$, all layers of v from layer 1 up to layer q are served by u_i itself, if it has already cached them (self-response). If u_i has not cached any or some layers of the demanded video quality, the remaining layers are fetched from the UEs in its UPN, eventually over multiple hops (UPN-response). In case none of the UEs in the UPN has cached the needed layers, the request is served by its associated SCBS (SCBS-response); otherwise, the layers are served by the MBS, thereby resulting in the highest delivery delay (MBS-response).

III. Problem Formulation

In this section, we formulate the problem of minimizing the delay of layered video caching (LVC), for the case where both the SCBSs and UEs have caching capabilities. Our objective is to determine the placement parameters $X=[X_s\ X_U]$ in such a way that the average video delivery delay is minimized. To achieve this goal, we first consider a case in which the video requests of each $u_i\in\mathcal{N}$ are served by the MBS M (MBS-response). In this case, the aggregated delivery delay of video requests of UEs can be derived as:

$$D_M^{NoCache}=\sum_{u_i\in\mathcal{N}}\sum_{v\in\mathcal{V}}\sum_{q\in\mathcal{Q}} p_v l_q \frac{o_{vq}}{c^w_{M,ui}} \tag{6}$$

where $c^w_{M,ui}$ denotes the achievable transmission rate between MBS M and user u_i. We assume that the transmission rate between M and u_i decreases from $c^w_M{=}c_1{+}c_2$ to c_1 ($c_1 > 0$), as the physical distance between them ($d_{M,ui}$) increases (e.g., due to the impact of path loss and interference).

We then consider the case where SCBS s caches the most popular videos, possibly allowing UEs to download the requested content directly from s with a lower delay than from M (SCBS-response). We again leverage file popularity to identify the cumulative probability that q-th quality level of a video cached by SCBS s is hit by a UE:

$$P^q_s = \frac{\sum_{k=1}^{x_{s,q}} k^{-\gamma}}{\sum_{j=1}^{V} j^{-\gamma}} \approx \frac{(x_{s,q})^{1-\gamma}}{(1-\gamma)\sum_{j=1}^{V} j^{-\gamma}} \tag{7}$$

where $x_{s,q}$ is the number of the most popular videos whose at least first q layers are cached by SCBS s. The rightmost term in Eq. (7) is derived by using a similar approximation as the one in [18]. Finally, videos requested by u_i could be found in its own cache (Self-response) or in the cache of the UEs in its UPN (UPN-response). Based on geographic caching (e.g., in [3]), the probability that there are n UEs within a distance r from a reference location for a PPP distribution with density λ is:

$$F(n,r,\lambda) = \frac{(\pi\lambda r^2)^n}{n!} e^{-\pi\lambda r^2} \tag{8}$$

Consequently, the probability that the q-th quality level of a video requested by u_i is found in its own cache or in the cache of at least one UE in the UPN of u_i within radius r_u is:

$$P^q_{ui} = 1{-}e^{-\pi\lambda_U r_u^2 p^q_{ui}} \tag{9}$$

where $e^{-\pi\lambda_U r_u^2 p^q_{ui}}$ is the probability that there is no UE within a distance r_u from u_i that cached the q-th quality level of the video. In Eq. (9), p^q_{ui} is derived as:

$$p^q_{ui} = \frac{(x_{ui,q})^{1-\gamma}}{(1-\gamma)\sum_{j=1}^{V} j^{-\gamma}} \tag{10}$$

Thus, the aggregated delivery delay of the requests of UEs served by the MBS M when some requested layers are served by SCBS s, UPN UEs, or each $u_{i\in N}$ is:

$$D^{Cache}_M = \sum_{u_i\in\mathcal{N}} \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}} (1-P^q_{ui})(1-P^q_s) p_v l_q \frac{o_{vq}}{c^w_{M,ui}} \tag{11}$$

Moreover, the aggregated delivery delay of the video requests of UEs served by SCBS s is:

$$D^{Cache}_s {=} \sum_{u_i\in\mathcal{N}} \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}} (1-P^q_{ui}) P^q_s p_v l_q \frac{o_{vq}}{c^w_s} \tag{12}$$

where P^q_{ui} and P^q_s are derived by using Eqs. (7) and (9), respectively. Here, c^w_s is characterized with respect to the physical distance between SCBS s and each $u_{i\in N}$.

Finally, the aggregated delivery delay of UEs' video requests served by the UE itself or its UPN is:

$$D^{UPN} {=} \sum_{u_i\in\mathcal{N}_s} \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}} P^q_{ui} p_v l_q \frac{o_{vq}}{c^f_{ui,uj}} \tag{13}$$

According to Eqs. (6) and (11)-(13), the delay saving when SCBS s and UEs participate in caching is given by:

$$\begin{aligned} D^{Saving} &= D^{NoCache}_M {-} D^{Cache}_M {-} D^{Cache}_s {-} D^{UPN} {=} \\ &\sum_{u_i\in\mathcal{N}} \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}} p_v l_q \frac{o_{vq}}{c^w_{M,ui}} - \\ &\sum_{u_i\in\mathcal{N}} \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}} (1{-}P^q_{ui})(1{-}P^q_s) p_v l_q \frac{o_{vq}}{c^w_{M,ui}} - \\ &\sum_{u_i\in\mathcal{N}} \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}} (1{-}P^q_{ui}) P^q_s p_v l_q \frac{o_{vq}}{c^w_{s,ui}} - \\ &\sum_{u_i\in\mathcal{N}} \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}} P^q_{ui} p_v l_q \frac{o_{vq}}{c^f_{ui,uj}} \end{aligned} \tag{14}$$

Consequently, maximizing the delay saving time in the LVC problem can be stated as the following integer linear programming problem:

$$\begin{aligned} \max_{X=[X_s\ X_U]} D^{Saving} {=} &\sum_{u_i\in\mathcal{N}} \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}} p_v l_q o_{vq} \Bigg(\frac{1}{c^w_{M,ui}} - \\ &\frac{(1{-}P^q_{ui})(1{-}P^q_s)}{c^w_{M,ui}} - \frac{(1{-}P^q_{ui})P^q_s}{c^w_{s,ui}} - \frac{P^q_{ui}}{c^f_{ui,uj}} \Bigg) \end{aligned} \tag{15}$$

$$\text{subject to} \sum_{j=1}^{Q} \sum_{k=1}^{n(s,j)} \sum_{q=1}^{j} o_{z(s,j)+k,q} \le c_s \tag{16}$$

$$\sum_{j=1}^{Q} \sum_{k=1}^{n(u_i,j)} \sum_{q=1}^{j} o_{z(u_i,j)+k,q} \le c_{ui}\ \forall i = 1,\ldots,|\mathcal{N}| \tag{17}$$

$$x_{s,Q} {\le}\ x_{s,Q-1} {\le} \ldots {\le} x_{s,1} {\le} V (x_{s,Q} {\ge} 0) \tag{18}$$

$$x_{i,Q} {\le}\ x_{i,Q-1} {\le} \ldots {\le} x_{i,1} {\le} V (x_{i,Q} {\ge} 0) \forall i = 1,\ldots,|\mathcal{N}| \tag{19}$$

which is equivalent to the following minimization problem:

$$\begin{aligned} \min_{X=[X_s\ X_U]} -D^{Saving} {=} - &\sum_{u_i\in\mathcal{N}} \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}} p_v l_q o_{vq} \Bigg(\frac{1}{c^w_{M,ui}} - \\ &\frac{(1{-}P^q_{ui})(1{-}P^q_s)}{c^w_{M,ui}} - \frac{(1{-}P^q_{ui})P^q_s}{c^w_{s,ui}} - \frac{P^q_{ui}}{c^f_{ui,uj}} \Bigg) \end{aligned} \tag{20}$$

subject to the constraints in Eqs. (16)-(19).

IV. Delay Optimization of Layered Video Caching

The LVC problem is an instance of the fractional knapsack problem [9], which is NP-hard. The $-D^{Saving}$ function in Eq. (20) is non-convex since its second order derivative (i.e., its Hessian matrix) is not positive definite. To find a solution to the problem, we employ an iterative approach, namely, the difference of convex (DC) functions programming [10]. The main idea in DC programming is to decompose a non-convex function into two convex functions in such a way that their combination is convex. Accordingly, we decompose function $-D^{Saving}$ into two convex functions $G(X)$ and $H(X)$ such that:

$$-D^{Saving}(X) {=} G(X) {-} H(X) \tag{21}$$

Algorithm 1 DC programming for the LVC problem

```
1 $k$=0 ▷ assign initial values
2 foreach $q \in Q$ do
3     $X^k_{s,q}=\frac{(Q-q+1)Vc_s}{QD_{size}}$
4     $X^k_{ui,q}=\frac{(Q-q+1)Vc_s}{QD_{size}}$ $(\forall i=1,\ldots,|\mathcal{N}|)$
5 while $\|X^k - X^{k+1} > \epsilon\|$ do
6     solve the following integer linear program:
        min $G(X){-}H(X^k){-}(X{-}X^k)\frac{\partial H(X^k)}{\partial X}$
        subject to Equations (16)-(19)
7     $k = k+1$
8 return $X^k$
```

Once the two functions $G(X)$ and $H(X)$ are found, a standard convex optimization method can be applied to solve function $G(X){-}H(X)$ instead of function $-D^{Saving}$ in Eq. (20). The main challenge in this step is to determine function $H(X)$ in such a way that both functions $H(X)$ and $G(X)$ are convex. We identify function $H(X)$ as follows:

$$H(X){=}\sum_{u_i\in\mathcal{N}}\sum_{v\in\mathcal{V}}\sum_{q\in\mathcal{Q}}\underbrace{p_v l_q o_{vq}\pi\lambda_U r_u^2 P_{ui}^{q^2}}_{h_{ivq}} \qquad (22)$$

The Hessian matrix of h_{ivq} is:

$$Hess(h_{ivq}){=}\begin{bmatrix}\frac{\partial^2 h_i}{\partial (P_s^q)^2} & \frac{\partial^2 h_i}{\partial P_s^q \partial P_{ui}^q}\\ \frac{\partial^2 h_i}{\partial P_{ui}^q \partial P_s^q} & \frac{\partial^2 h_i}{\partial (P_{ui}^q)^2}\end{bmatrix}=\begin{bmatrix}0 & 0\\ 0 & 2p_v l_q o_{vq}\pi\lambda_U r_u^2\end{bmatrix}$$

which is positive definite and $H(X)$ is convex in X. Similarly, the Hessian matrix of $G(X){=}H(X){-}D^{Saving}$ is positive definite and $G(X)$ is convex. Thus, we can consider $G(X){-}H(X)$ instead of $-D^{Saving}$ with the same constraints in Eqs. (16)-(19). Accordingly, we design Algorithm 1 to find the values of X in an iterative manner. First, we assign the initial values of X (lines 1-4), where D_{size} is the size of the dataset. Even though the initial values can be assigned at random, identifying suitable starting points can accelerate convergence. Next, we solve the integer linear programming problem in line 6 until $X^k{-}X^{k+1}$ converges (lines 5-7).

V. Numerical Results

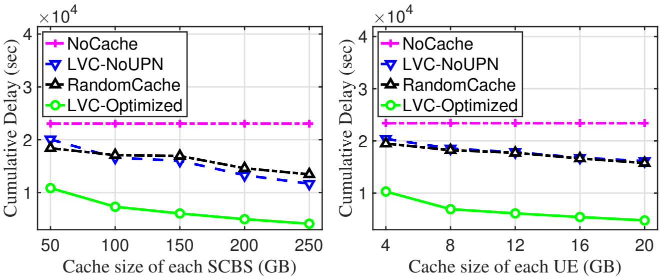
We now present numerical results obtained by simulation to evaluate the performance of our proposed caching scheme. We consider a wireless network consisting of one MBS located at the center of a circular area, as in Fig. 1. The network includes 4 SCBSs and 300 UEs distributed according a PPP. The parameters employed in the simulation are detailed in Table II. We use the real video dataset in [1], with a total size of about 1 TB, to generate layered videos. We run 10 replications of the experiments, each lasting for $T = 1,000$ time slots of one second. The results show the resulting average values; the standard deviations were very small, thus, we did not report them in the plots for the sake of readability.

We consider four different schemes: NoCache in which the requests of UEs are only served by the MBS (i.e., no caching takes place in the network); LVC-NoUPN in which video layers are cached at the SCBSs based on our solution to the LVC problem, and the requests of each UE are served only by either its associated SCBS or the MBS; RandomCache in which video layers at both the SCBSs and UEs are selected at random, and the requests of each UE are served by its UPN, its associated SCBS, or the MBS; LVC-Optimized in which video layers are determined based on our solution to the LVC problem, and the requests of each UE are served by its UPN, its associated SCBS, or the MBS. In the following, we focus on the *cumulative delay*, namely, the total time it takes for UEs to download all the requested videos.

TABLE II: Simulation parameters

Parameter	**Value**
Transmission range of MBS M: r_M	300 (m)
Transmission capacity of MBS M: c^w_M	1 (Mbps)
Number of SCBSs: $\|\mathcal{S}\|$	4
Transmission range of each SCBS: r_s	80 (m)
Storage capacity of each SCBS: c_s	100 (GB)
Number of UEs: $\|U\|$	300
Poisson point process (PPP) density: λ_U	$300/\pi 300^2{=}0.001$
Communication range of each UE: r_u	30 (m)
Storage capacity of each UE: c_{ui}	2∼5 (GB)
Max. LTE-A downloading capacity: c^w_{ui}	12 (Mbps) [8]
Max. Wi-Fi downloading capacity: c^f_{ui}	4 (Mbps) [8]
Max. Wi-Fi downloading between UEs: $c^f_{ui,uj}$	64 (Mbps) [8]
Number of videos: $\|\mathcal{V}\|$	5000
Number of video layers: Q	5
Demand rate of each UE: ζ_u	0.2 (per time slot)
Skewness of Zipf distribution: γ	0.8 [14]

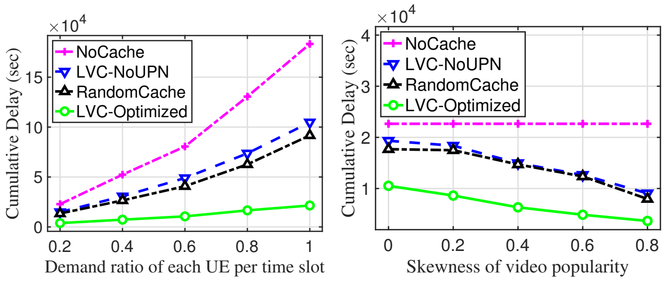
Fig. 3 shows the cumulative delay for the different schemes as a function of the cache size of SCBSs and UEs. In particular, Fig. 3a shows that the cumulative delay for all schemes excluding NoCache decreases as the cache size of SCBS increases, whereas the delay of NoCache remains the same. The figure also demonstrates that the cumulative delay is considerably lower for LVC-Optimized compared to the other schemes. For instance, when the cache size of SCBSs is 150 GB, LVC-Optimized outperforms the LVC-NoUPN, RandomCache and NoCache schemes by 90%, 95%, and



(a)

(b)

Fig. 3: Cumulative delay as a function of the cache size of (a) SCBSs and (b) UEs.



(a) (b)

Fig. 4: Cumulative delay as a function of (a) the demand ratio of UEs and (b) the skewness of video popularity.

116%, respectively. The main reason is that LVC-Optimized caches the most popular video layers in SCBSs, which increases the cache hit probability, resulting in a lower delay. Moreover, Fig. 3a shows that the delay of LVC-NoUPN is higher than that of RandomCache when the cache size of SCBSs is less than about 100 GB, whereas LVC-NoUPN outperforms RandomCache when the cache size of SCBSs exceeds 100 GB. The reason is that the cache hit ratio in SCBSs is higher for LVC-NoUPN, because the SCBSs in this case caches the most popular videos, while SCBSs and UEs in RandomCache cache the videos randomly. The trends in Fig. 3b are similar to those Fig. 3a for all the scenarios, as the cache size of UEs increases.

Fig. 4a illustrates the cumulative delay as a function of the video demand ratio for the different schemes. The figure clearly shows how the delay increases as the demand ratio of UEs increases too. While the delay of LVC-Optimized increases gradually over the considered range, the other schemes exhibit a much sharper increase when the demand ratio exceeds 0.6. Specifically, when the demand ratio is 0.8, the delay in LVC-Optimized is 154%, 126%, and 116% less than NoCache, LVC-NoUPN, and RandomCache, respectively. The reason is that, in contrast to the other schemes, a large portion of the video requests of UEs in LVC-Optimized are served through the layers cached at the SCBSs or UPN.

Finally, Fig. 4b illustrates the cumulative delay as a function of the skewness in the video popularity for the different schemes. Clearly, the download delay of UEs in all schemes except NoCache decreases as the skewness increases. However, the delay in LVC-Optimized is considerably lower than that of other schemes. For instance, when the skewness parameter is 0.6, the cumulative delay of UEs in LVC-Optimized is 129%, 88%, and 87% less than the NoCache, LVC-NoUPN, and RandomCache, respectively. The reason is that LVC-Optimized caches the most popular videos at the UEs and SCBSs, resulting in a higher number of requests that can be served through cached layers. This happens because UEs request popular videos more frequently than others.

VI. Conclusion

In this article, we analyzed the delay of layered video caching and delivery in crowdsourced HetNets. Our main objective was to identify the best layers to be cached at SCBSs and UEs based on their popularity, so as to minimize the video download time. Since delay minimization in layered video caching is NP-hard, we employed DC programming and obtained the set of video layers to be cached in an iterative manner. Our evaluation showed that caching more layers of the most popular videos significantly decreases the video download time. Moreover, cooperative caching and delivery of video layers through UPN considerably reduces the average download delay too. As a future work, we plan on addressing the energy-delay trade-off of layered video caching.

Acknowledgments

This work was partially supported by the Academy of Finland grants number 299222 and number 305507. We would like to thank Sergiy Vorobyov for his comments.

References

[1] Video trace library. http://trace.eas.asu.edu/.

[2] E. Bastug, M. Bennis, and M. Debbah. Living on the edge: The role of proactive caching in 5G wireless networks. *IEEE Communications Magazine*, 52(8):82–89, Aug 2014.

[3] B. Blaszczyszyn and A. Giovanidis. Optimal geographic caching in cellular networks. In *Proc. IEEE ICC*, pages 3358–3363, June 2015.

[4] Z. Chen, N. Pappas, and M. Kountouris. Probabilistic caching in wireless D2D networks: Cache hit optimal versus throughput optimal. *IEEE Communications Letters*, 21(3):584–587, March 2017.

[5] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021. 2016, tech. rep.

[6] Cisco Webcasts. Emerging video technologies: H.265, SVC, and WebRTC. 2014, [Online]. Available: https://www.ciscolive.com.

[7] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire. Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution. *IEEE Communications Magazine*, 51(4):142–149, April 2013.

[8] Junxian Huang, Feng Qian, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. A close examination of performance and power characteristics of 4G LTE networks. In *Proc. ACM MobiSys '12*, pages 225–238, 2012.

[9] Hiroaki Ishii, Toshihide Ibaraki, and Hisashi Mine. Fractional knapsack problems. *Mathematical Programming*, 13(1):255–271, Dec 1977.

[10] Hoai An Le Thi, Van Ngai Huynh, and Tao Pham Dinh. DC programming and DCA for general DC programs. In Tien van Do, Hoai An Le Thi, and Ngoc Thanh Nguyen, editors, *Advanced Computational Methods for Knowledge Engineering*, pages 15–35, 2014.

[11] L. Li, G. Zhao, and R. S. Blum. A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies. *IEEE Comm. Surveys and Tutorials*, 2018.

[12] W. Li, S. M. A. Oteafy, and H. S. Hassanein. On the performance of adaptive video caching over information-centric networks. In *Proc. IEEE ICC*, pages 1–6, 2017.

[13] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu. Understanding performance of edge content caching for mobile video streaming. *IEEE Journal on Selected Areas in Communications*, 35(5):1076–1089, May 2017.

[14] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas. Caching and operator cooperation policies for layered video content delivery. In *Proc. IEEE INFOCOM*, pages 1–9, April 2016.

[15] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, Sept 2007.

[16] D. Syrivelis, G. Iosifidis, D. Delimpasis, K. Chounos, T. Korakis, and L. Tassiulas. Bits and coins: Supporting collaborative consumption of mobile internet. In *Proc. IEEE INFOCOM*, pages 2146–2154, April 2015.

[17] Y. Wang and X. Lin. CHetNet: crowdsourcing to heterogeneous cellular networks. *IEEE Network*, 29(6):62–67, Nov 2015.

[18] J. Xie, R. Xie, T. Huang, J. Liu, and Y. Liu. Energy-efficient content placement for layered video content delivery over cellular networks. In *Proc. IEEE GLOBECOM*, pages 1–6, Dec 2017.