# Energy Efficiency and Delay Optimization for Edge Caching Aided Video Streaming

Guorong Zhou, Liqiang Zhao, *Member, IEEE*, Yunfeng Wang, Gan Zheng, *Senior Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—In this paper, we design a computing, communication and caching scheme for edge caching-based video streaming in order to improve the network performance. Firstly, we optimize the system's energy efficiency and delay with the aid of network function virtualization. Then, a dynamic edge caching decision is developed, and based on Lyapunov optimization, an alternating resource optimization algorithm is proposed for allocating the optimal subcarrier and power resources, video caching and computing resources. Our numerical results show that the proposed scheme outperforms both the traditional caching scheme as well as the least frequently used (LFU)-40% regime, and strikes a compelling tradeoff between the energy efficiency and delay.**

***Index Terms*—edge caching, video streaming, computing, energy efficiency, delay.**

## I. Introduction

For high-throughput video streaming, edge caching substantially reduces users' delay, whilst additionally improving the network performance [1]. In traditional caching at the edge, the popular videos will be completely stored in advance in the edge and then transmitted without relying on the backhaul link when requested [2].

The literature of edge caching has evolved rapidly [3]–[5]. For a given cache memory budget, Zhang *et al.* [3] optimized the cache size of macro and small base stations (BSs) in heterogeneous networks for maximizing the overall network capacity. Ma *et al.* [4] proposed to exploit both the temporal and spatial video request patterns observed for improving the performance by edge content caching, while Li *et al.* [5] minimized the average video distortion of all users. They all improve the user experience by caching strategies or by optimizing the edge cache size. However, the storage space in the edge is always limited. In order to circumvent this problem, video compression is introduced, which only caches the compressed videos and the corresponding transcoding parameters at the edge [6]. When requested, the compressed video can be transcoded into different-resolution versions required by the users on the fly.

In this way, video compression can alleviate the shortage of edge storage space and congestion of the backhaul link through online transcoding, so as to integrate computing, caching, and communication (3C) resources. However, existing contributions only manage one or two of the resources [7], [8]. Explicitly, Sun *et al.* [7] proposed a hierarchical wireless resource allocation architecture, in which subchannels are first allocated to the local resource managers and then to the users. Wang *et al.* [8] considered the offloading decision and resource allocation in the twin-fold context of communication and computing.

Nevertheless, the 3C resources exploited for optimizing the energy efficiency (EE) of video streaming are characterized in this treatise for the first time, where the most popular videos are cached without compression, while the remaining videos are compressed and then cached for enhancing the performance of vehicular networks. Specifically, we formulate the problem of maximizing the network's EE, while satisfying the delay constraints by developing a near-instantaneously adaptive edge caching (NAEC) decision regime and optimizing both the subcarrier and power allocation, as well as the computing resources. Then by invoking Lyapunov optimization, an alternating resource optimization (ARO) algorithm is proposed for solving the above problem. Finally, our numerical results show that the proposed 3C scheme outperforms both the traditional caching scheme as well as the LFU-40% regime, and the associated tradeoff between the EE and delay is characterized. The main contributions and our comparisons to the relevant references are shown in Table 1.

## II. System Model

For simplicity, we assume that a single physical base station (BS) supports $K$ users and the BS has certain computing and caching capability. With the aid of network function virtualization (NFV), the physical BS could be virtualized into a pair of virtual BSs (vBSs), e.g., vBS1 and vBS2, where vBS1 is equipped with a certain storage space for caching uncompressed videos, while vBS2 has sufficient computing resources used to transcode compressed videos, as shown in Fig. 1. For convenience of analysis, we partition the different-size and different-resolution video files into video blocks of the same size. Assume that there are $L$ video blocks, each

Guorong Zhou, Liqiang Zhao, Yunfeng Wang (e-mail: guor_zhou@163.com, lqzhao@mail.xidian.edu.cn, wyf1905@qq.com) are with the State Key Laboratory of Integrated Service Networks at Xidian University, Xi'an 710071, China. Gan Zheng (e-mail: g.zheng@lboro.ac.uk) is with the Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, UK. Lajos Hanzo (e-mail: lh@ecs.soton.ac.uk) is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK.

This work was supported in part by National Natural Science Foundation of China (61771358, 61901317), Fundamental Research Funds for the Central Universities (JB190104), Science and Technology Plan of XiAn City (2019217014GXRC006CG007-GXYD6.1), National Key Research and Development Project (2017YFE0121300), and the 111 Project (B08038). G. Zheng would like to acknowledge the UK EPSRC under grant number EP/N007840/1 and the Leverhulme Trust Research Project Grant under grant number RPG-2017-129. L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/N004558/1, EP/P034284/1, EP/P034284/1, EP/P003990/1 (COALESCE), of the Royal Society's Global Challenges Research Fund Grant as well as of the European Research Council's Advanced Fellow Grant QuantCom.

Table 1: Our Main Contributions.

| Novelty | [2] | [3] | [4] | [5] | [6] | [7] | [8] | Our paper |
|---|---|---|---|---|---|---|---|---|
| Video streaming scenario | | | ✓ | ✓ | ✓ | | | ✓ |
| Theory of video compression (transcoding online) | | | | | ✓ | | | ✓ |
| Theory of edge caching | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ |
| Dynamic edge caching decision | | | ✓ | | ✓ | | | ✓ |
| Allocation of computing resource | | | | | ✓ | | ✓ | ✓ |
| Allocation of communication resource (subcarrier) | | | | | | ✓ | ✓ | ✓ |
| Allocation of communication resource (power) | | | | | | ✓ | ✓ | ✓ |
| Optimization of network performance (EE) | ✓ | | | | | | ✓ | ✓ |
| Optimization of network performance (delay) | | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tradeoff between EE and delay | | | | | | | | ✓ |

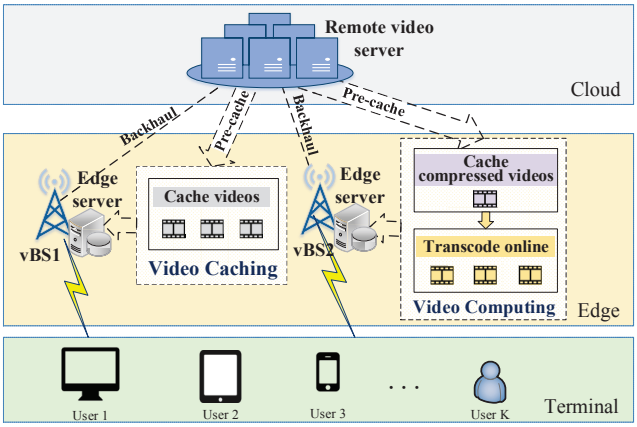


Figure 1: System model.

having $M$ bits and an index set of $\boldsymbol{L}=\{1,...,l,...,L\}$. The vBS1 can cache $C_n$ video blocks, where $C_n<L$. The system has time slots $t\in\{0,1,2,...,T\}$ of duration $\tau$. The procedure of edge caching-based video streaming is divided into two phases: the pre-caching phase during off-peak hours and the online transmission phase.

### A. Pre-caching Phase

In the pre-caching phase, we propose a NAEC decision based on least frequently used (LFU) strategy, which means video blocks in vBS1 that are least frequently used in a certain time period are removed when full [9]. The mobile network operator (MNO) stores either uncompressed or compressed video blocks at the edge according to their popularity. Assume that the system does not know the time-varying popularity in advance. Therefore, the popularity of video blocks for user $k,\forall k\in K$ may be determined from the user's historical requests during the previous $T_1$ transmission slots ($T_1<T$), and then some of the cached videos may be replaced depending on the predicted popularity based on the previous request history.

Without loss of generality, we use the Zipf distribution to indicate the popularity probability of video block $l$ for user $k$, which is formulated as $q_{k,l}(t)=\frac{[O_{k,l}(t)]^{-\alpha}}{\sum_{i=1}^{L}[O_{k,i}(t)]^{-\alpha}}$, where $O_{k,l}(t)$ is the popularity order of video block $l$ for user $k$ arranged in descending sequence based on the user's historical requests. In other words, $\alpha$ indicates the degree of popularity, which is typically in the range of $\alpha\in[0.5,1.5]$. A large $\alpha$ represents more requests concerning popular video blocks and less requests for unpopular ones. Let $q_l(t)=\frac{1}{K}\sum_{k=1}^{K}q_{k,l}(t)$ denote the average popularity probability of video block $l$ requested by all users in the whole network. And the full set is given by $\boldsymbol{q}(t)=\{q_1(t),...,q_l(t),...,q_L(t)\}$, while its sorted version arranged in descending order of $\boldsymbol{q}(t)$ is formulated as $\boldsymbol{O}(t)=\Pi(\boldsymbol{q}(t))$.

Therefore, vBS1 stores the top $C_n$ video blocks of $\boldsymbol{O}(t)$, while vBS2 stores the remaining $(L-C_n)$ compressed video blocks and transcoding parameters. To facilitate our mathematical analysis, we neglect the compressed video's storage space requirement at the edge in this paper. The popularity order $O_{k,l}(t)$ of a video block of user $k$ will be updated once every $T_1$ transmission slots, so will $\boldsymbol{O}(t)$. The specific steps of the NAEC decision in pre-caching phase are summarized in Algorithm 1. It can be found that the number of video blocks evicted and replaced in both vBSs varies in every update.

### B. Transmission Phase

During the transmission phase, the MNO assigns the most suitable vBS to users. If the requested contents have been cached without compression, the user is associated with vBS1 directly for video-acquisition; otherwise the user is associated with vBS2 for fetching, transcoding and transmission. Therefore, we divide this phase into two processes: computing process and transmission process, which are described below, respectively.

*1) **Computing Process**:* Let us assume that the set of video blocks requested by user $k$ at slot $t$ is $\boldsymbol{d_k}(t)=\{d_{k,1}(t),...,d_{k,n}(t)\}$ and the total number is $n$. The number of compressed video blocks is $\sum_{l=1}^{n}y(\Pi[d_{k,l}(t)])$, where $\Pi[d_{k,l}(t)]$ is the new position of $d_{k,l}(t)$ according to $\Pi(\boldsymbol{q}(t))$, and $y(i)=1$ if $i>C_n$, otherwise $y(i)=0$. Then the load that has to be computed for user $k$ at slot $t$ is $A_k^c(t)=M\cdot\sum_{l=1}^{n}y(\Pi[d_{k,l}(t)])$, so we model the queuing process of computing for user $k$ as:

$$Z_k(t+1)=\max\{Z_k(t)-[f_k(t)/c_k]\tau+A_k^c(t),0\},\forall k, \quad (1)$$

where $[f_k(t)/c_k]$ accounts for the local computing rate of user $k$ at a computing clock frequency of $f_k(t)$ expressed in CPU cycle/s, and $c_k$ is the number of computing cycles required per bit in CPU cycle/bit [10]. The network is said to be stable, when all the $K$ queues are mean-rate stable [11]. We define $\boldsymbol{F}(t)=\{f_k(t)\}$ as the set of computing clock frequency. Meanwhile, we have the time-averaged expectation of $f_k(t)$ as

$$\overline{f}_k=\lim_{I\to\infty}\frac{1}{I}\sum_{t=0}^{I-1}E\{f_k(t)\}. \quad (2)$$

Given $f_k(t)$, the power consumption of computing can be calculated as $r[f_k(t)]^3$ in which the parameter $r$ depends on the hardware architecture [10]. Therefore, based on $E=Pt$, which indicates that energy consumption is equal to the

**Algorithm 1** Near-instantaneously Adaptive Edge Caching (NAEC) Decision in Pre-caching Phase.

1: **Input:** The set $\boldsymbol{d_k}(t)$ of video blocks requested by user $k$ at slot $t$.
2: **Output:** $\boldsymbol{O}(t)$, and update cached video blocks in vBS1 and vBS2.
3: Set $t = mT_1, m \in \mathbf{N}$.
4: Initialization: When $m = 0$, vBS1 caches the first $C_n$ uncompressed video blocks according to users' request order, and vBS2 caches the remaining compressed video blocks; $\boldsymbol{O}(t) = \emptyset$.
5: **REPEAT**
6: Let $m = m + 1$.
7: Obtain popularity order of $O_{k,l}(t)$ during the previous $T_1$ transmission slots, and its probability $q_{k,l}(t)$ by Zipf distribution.
8: Obtain the full set $\boldsymbol{q}(t)$ of the average popularity probability in the whole network.
9: Obtain the sorted version $\mathbf{O}(t) = \Pi(\mathbf{q}(t))$ arranged in descending order of $\boldsymbol{q}(t)$.
10: Compare the existing video blocks stored in vBS1 with the top $C_n$ video blocks in $\boldsymbol{O}(t)$. The same are reserved, and different are evicted and replaced by LFU. Then use the same method to evict and replace compressed videos in vBS2.
11: **STOP** when $t \geq T$.

product of power and time, all users' computing-related energy consumption $E_{comp}(t)$ in slot $t$ is given by

$$E_{comp}(t) = \sum\nolimits_{k=1}^{K} r[f_k(t)]^3 \cdot [A_k^c(t)\, c_k]/f_k(t). \tag{3}$$

We then have the time-averaged expectation of $E_{comp}(t)$ as

$$\overline{E}_{comp} = \lim_{I\to\infty} \frac{1}{I} \sum\nolimits_{t=0}^{I-1} E\{E_{comp}(t)\}. \tag{4}$$

*2) **Transmission Process**:* Regardless whether associated with vBS1 or vBS2, all video blocks requested have to be transmitted over wireless channels. Let us consider the downlink (DL) transmission in an orthogonal frequency division multiple access (OFDMA) system, where no inter-user interference is encountered. The DL bandwidth $B$ Hz is partitioned into $S$ subcarriers and each subcarrier has a bandwidth of $B/S$ Hz. The channel coefficient $h_{k,s}(t)$ of user $k$ on subcarrier $s$ ($\forall s \in S$) at slot $t$ is independent and identically distributed (i.i.d.) over time and $p_{k,s}(t)$ is the transmit power of user $k$ on subcarrier $s$ at slot $t$, while $\boldsymbol{P}(t) = \{p_{k,s}(t)\}$ is the set of transmit powers. The achievable data rate $r_{k,s}(t)$ of user $k$ on subcarrier $s$ at slot $t$ is given by

$$r_{k,s}(t) = \frac{B}{S}\log_2\left[1 + \frac{p_{k,s}(t)\,|h_{k,s}(t)|^2}{(\frac{B}{S})N_0}\right], \tag{5}$$

where the power spectral density (PSD) of the additive white Gaussian noise (AWGN) is $N_0$. We define $\boldsymbol{X}(t) = \{x_{k,s}(t)\}$ as the set of subcarrier allocation indicators and $x_{k,s}(t)$ is a time-sharing factor of user $k$ on subcarrier $s$. Based on (5), the total DL transmission rate of user $k$ at slot $t$ is given by

$$R_k(t) = \sum\nolimits_{s=1}^{S} x_{k,s}(t) r_{k,s}(t), \forall k. \tag{6}$$

Similar to (2), we have the time-averaged expectation of $\overline{R}_k = \lim_{I\to\infty} \frac{1}{I}\sum_{t=0}^{I-1} E\{R_k(t)\}$. Moreover, the DL transmission power consumption of user $k$ at slot $t$ is modeled as

$$P_k(t) = \sum\nolimits_{s=1}^{S} x_{k,s}(t) p_{k,s}(t) + P_k^c, \forall k, \tag{7}$$

where $P_k^c$ denotes the circuit power consumption for user $k$. Then the time-averaged expectation is defined as $\overline{P}_k = \lim_{I\to\infty} \frac{1}{I}\sum_{t=0}^{I-1} E\{P_k(t)\}$.

The traffic load required by user $k$ for wireless transmission at slot $t$ is $A_k^a(t) = M \cdot n$. Thus, we model the queuing process of transmission for user $k$ by:

$$Q_k(t+1) = \max\{Q_k(t) - R_k(t)\,\tau + A_k^a(t), 0\}, \forall k. \tag{8}$$

Without loss of generality, we assume the total delay to be the sum of transmission queuing delay and computing queuing delay. Due to the fact that the user's average queuing delay is proportional to the average queue length according to Little's Theorem [12], we can represent the delay of user $k$ by the sum of the queue length $[Q_k(t) + Z_k(t)]$. In addition, by $E = Pt$, we get the energy consumption of wireless transmission $E_{access}(t)$ in slot $t$ as:

$$E_{access}(t) = \sum\nolimits_{k=1}^{K} P_k(t) \cdot \frac{A_k^a(t)}{R_k(t)} = \sum\nolimits_{k=1}^{K} \frac{P_k(t)\, Mn}{R_k(t)}. \tag{9}$$

The time-averaged expectation of $E_{access}(t)$ is given by

$$\overline{E}_{access} = \lim_{I\to\infty} \frac{1}{I} \sum\nolimits_{t=0}^{I-1} E\{E_{access}(t)\}. \tag{10}$$

Let us now analyze the EE performance of the 3C scheme. The time-averaged network EE is deduced from the definition of EE as the ratio of total number of requested bits to the total energy consumption, as follows:

$$\overline{U}_{EE}(\boldsymbol{X}, \boldsymbol{P}, \boldsymbol{F}) = \frac{nMK}{\overline{E}_{comp} + \overline{E}_{access}}. \tag{11}$$

## III. Problem Formulation and solution

We aim for maximizing the network's EE of the proposed scheme, while satisfying the user's delay constraint including the transmission and computing delay in the edge caching-based video streaming service, i.e.,

$$\begin{aligned}
&\min_{\boldsymbol{X},\boldsymbol{P},\boldsymbol{F}} \frac{1}{\overline{U}_{EE}(\boldsymbol{X}, \boldsymbol{P}, \boldsymbol{F})}, \\
&\text{s.t. } C1: Q_k(t) \leq \beta, \forall k, t, \qquad C2: Z_k(t) \leq \omega, \forall k, t, \\
&\quad C3: \bar{R}_k \geq R_k^{av}, \forall k, \qquad C4: \overline{f}_k \leq f_k^{\max}, \forall k, \\
&\quad C5: f_k(t) \geq 0, \forall k, t, \qquad C6: 0 \leq x_{k,s}(t) \leq 1, \forall k, s, t, \\
&\quad C7: \sum\nolimits_{k=1}^{K} x_{k,s}(t) \leq 1, \forall s, t, \; C8: P_k(t) \leq P_k^{\max}, \forall k, t, \\
&\quad C9: p_{k,s}(t) > 0, \forall k, s, t,
\end{aligned} \tag{12}$$

where $R_k^{av}$, $f_k^{max}$ and $P_k^{max}$ are the average rate requirement, the maximum computing capability and power of user $k$, respectively. Furthermore, $C1$ and $C2$ guarantee the queueing stability under the maximum queue lengths of $\beta$ and $\omega$, respectively, while $C3$ guarantees the average rate requirements of user $k$. Additionally, $C6$ and $C7$ denote the time sharing constraints of subcarriers, while $C4$, $C5$, $C8$ and $C9$ are the peak and nonnegative computing frequency constraints, as well as the peak and nonnegative transmission power constraints, respectively.

To reduce the complexity in (12), we propose an ARO algorithm by separating the edge caching subproblem from the communication subproblem to make the optimization problem easier to solve.

### A. Communication Subproblem

For the subcarrier and power allocation in the communication subproblem, we have

$$\begin{aligned}
&\min_{\boldsymbol{X},\boldsymbol{P}} \overline{E}_{access}, \\
&\text{s.t. } C1, C3, C6 - C9.
\end{aligned} \tag{13}$$

Using the generalized fractional programming theory of [13] and a similar transform to the stochastic optimization problem of [14], the nonlinear fractional problem of (13) is transformed into

$$\begin{aligned}
&\min \sum\nolimits_{k=1}^{K} [\overline{P}_k(\boldsymbol{X}, \boldsymbol{P}) - \pi_k^{EE}(t)\overline{R}_k(\boldsymbol{X}, \boldsymbol{P})], \\
&\text{s.t. } C1, C3, C6 - C9,
\end{aligned} \tag{14}$$

where $\pi_k^{EE}(t) = \frac{\sum_{\tau=0}^{t-1} P_k[\boldsymbol{X}(\tau), P(\tau)]}{\sum_{\tau=0}^{t-1} R_k[\boldsymbol{X}(\tau), P(\tau)]}$.

**Algorithm 2** Alternating Resource Optimization (ARO) Algorithm to Solve (12).

1: Initialization:$Q_k(0)$=0,$Z_k(0)$=0,$G_k(0)$=0,$Y_k(0)$=0,and$\pi_k^{EE}(0)$=0.
2: **REPEAT**
3: Set $d=0$. Initialize the lagrange multipliers $\theta_k$, the step size $e^d$.
4: **repeat**
Solve the power allocation, subcarrier allocation by using (20) and (21), respectively.
$d=d+1$; Update $\theta_k^d$ by subgradient method.
**until** Convergence = **true** or $d>d^{\max}$
**return** $\{\boldsymbol{X}^*(t),\boldsymbol{P}^*(t)\}$ as the optimal result.
5: Update user's rate $R_k(t)$ and power $P_k(t)$ by (6) and (7).
6: When $t=mT_1, m\in\mathbf{N}$, update the video popularity set $\boldsymbol{O}(t)$ and cached video blocks in vBS1 and vBS2 by Algorithm 1.
7: Update $\boldsymbol{F}^*(t)$ according to (28).
8: Let $t=t+1$.
9: Update $Q_k(t)$, $Z_k(t)$, $G_k(t)$, $Y_k(t)$ and $\pi_{EE}(t)$ according to (15), (25), (16), (26) and (14), respectively.
10: **STOP** when $t=T$.

We transform the transmission queue according to $C1$ into:

$$Q_k(t+1)=\begin{cases}\max[Q_k(t)-R_k(t)\tau,0], if\ Q_k(t)>\beta,\\ \max[Q_k(t)-R_k(t)\tau+A_k^a(t),0], otherwise.\end{cases} \tag{15}$$

The queue is denoted as $\mathcal{Q}(t)=\{Q_k(t)\}$.

Based on the general Lyapunov theory of [11], we transform $C3$ in (12) into a virtual rate queue stability problem for simplifying it. We denote the virtual rate queues as $\mathcal{G}(t)=\{G_k(t)\}$, which are updated as follows:

$$G_k(t+1)=\max[G_k(t)+R_k^{av}\tau-R_k(t)\tau,0],\forall k,t, \tag{16}$$

where $G_k(0)=0$. The vector $\Theta(t)=\left[\mathcal{Q}(t),\mathcal{G}(t)\right]$ is defined to represent the queuing states of all queues. According to [11], the drift-plus-penalty function $df(t)$ is given by

$$\begin{aligned}df(t)&=E\{L[\Theta(t+1)]-L[\Theta(t)]|\Theta(t)\}\\&+V\mathbb{E}\{\sum_{k=1}^K[\overline{P}_k(\boldsymbol{X},\boldsymbol{P})-\pi_k^{EE}(t)\overline{R}_k(\boldsymbol{X},\boldsymbol{P})]\},\end{aligned} \tag{17}$$

where the Lyapunov function is defined as

$$L(\Theta(t))\triangleq\sum_{k=1}^K\{[Q_k^2(t)+G_k^2(t)]/2\}, \tag{18}$$

and $V\geq 0$ is a control parameter invoked for striking a tradeoff between the delay and EE.

According to stochastic optimization theory [11], the optimal solution of the problem (14) at slot $t$ can be obtained by minimizing the upper bound of $df(t)$ as follows

$$\begin{aligned}&\min_{\boldsymbol{X},\boldsymbol{P}}\sum_{k=1}^K\{\xi_1Q_k(t)A_k^a(t)-\tau Q_k(t)R_k(t)+\\&\tau G_k(t)[R_k^{av}-R_k(t)]\}+V\sum_{k=1}^K\{P_k(t)-\pi_k^{EE}(t)R_k(t)\},\\&\text{s.t.}\quad C6-C9,\end{aligned} \tag{19}$$

where $\xi_1$ is a binary factor, which equals to 1 if the queue length is below the transmission queue length threshold $\beta$ and 0 otherwise.

Following a similar approach presented in Section 3.2 of [13], we can prove that (19) is convex and its closed-form solutions can be obtained by the Karush-Kuhn-Tucker (KKT) conditions. We obtain the optimal power allocation policies as

$$p_{k,s}^*(t)=\left[\frac{B}{S}\left\{\frac{\pi_k^{EE}(t)+[\tau Q_k(t)+\tau G_k(t)]/V}{(1+\theta_k/V)\ln 2}-\frac{N_0}{|h_{k,s}(t)|^2}\right\}\right]^+, \tag{20}$$

where $[y]^+=\max[y,0]$, and $\theta_k$ is the Lagrange multiplier corresponding to constraint $C8$. Substituting (20) into (19), we obtain the optimal subcarrier assignment as

$$x_{k,s}^*(t)=\begin{cases}1, if\ \varphi_{k,s}(t)<0 \text{ and } k=\arg\min\limits_{1\leq k\leq K}\varphi_{k,s}(t),\\ 0, \text{otherwise},\end{cases} \tag{21}$$

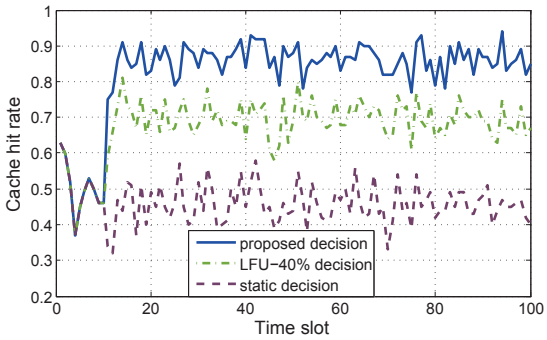


Figure 2: Cache hit rate comparison ($T_1=10$ time slots).

where

$$\begin{aligned}\varphi_{k,s}(t)&=(V+\theta_k)\,p_{k,s}^*(t)-\left[\tau Q_k(t)+\tau G_k(t)+V\pi_k^{EE}(t)\right]\\&\times\frac{B}{S}\log_2\left\{1+\frac{p_{k,s}^*(t)|h_{k,s}(t)|^2}{(B/S)\,N_0}\right\}.\end{aligned} \tag{22}$$

Using the sub-gradient-based method, the update of $\theta_k$ is given as

$$\theta_k^{(d+1)}(t)=\left[\theta_k^d(t)-e^d\left\{P_k^{\max}-\left[\sum_{s=1}^S x_{k,s}^*(t)p_{k,s}^*(t)+P_k^c\right]\right\}\right]^+, \tag{23}$$

where $d$ denotes the iteration index having the maximum value of $d^{\max}$ and $e^d$ is the step size. For sufficiently small $e^d$, the primal variable $p_{k,s}(t)$ and $x_{k,s}(t)$ will converge to the optimal $p_{k,s}^*(t)$ and $x_{k,s}^*(t)$ as $d\to\infty$, respectively [15].

### *B. Edge Caching Subproblem*

For the computing resource allocation of our edge caching subproblem, we have

$$\begin{aligned}&\min_{\boldsymbol{F}}\overline{E}_{comp},\\&\text{s.t.}\quad C2,C4,C5.\end{aligned} \tag{24}$$

Similar to the derivation of (13), based on Lyapunov theory [11], we transform $C2$ and $C4$ into, respectively:

$$Z_k(t+1)=\begin{cases}\max[Z_k(t)-[f_k(t)/c_k]\tau,0], if\ Z_k(t)>\omega,\\ \max[Z_k(t)-[f_k(t)/c_k]\tau+A_k^c(t),0], otherwise,\end{cases} \tag{25}$$

$$Y_k(t+1)=\max[Y_k(t)+f_k(t)\tau-f_k^{max}(t)\tau,0],\forall k,t. \tag{26}$$

Then the optimal subproblem (24) may be transformed into:

$$\begin{aligned}&\min_{\boldsymbol{F}}\sum_{k=1}^K\left\{Vr[f_k(t)]^2c_k\sum_{l=1}^n y(\Pi(d_{k,l}(t)))\right.\\&\left.+\xi_2Z_k(t)A_k^c(t)-\tau Z_k(t)[f_k(t)/c_k]+\tau Y_k(t)[f_k(t)-f_k^{\max}]\right\},\\&\text{s.t.}\quad C5,\end{aligned} \tag{27}$$

where $\xi_2$ is a binary factor, which equals to 1 if the queue length is below the threshold $\omega$ and 0 otherwise. Specifically, according to Algorithm 1, we can evaluate the video popularity set $\boldsymbol{O}(t)$ and update cached video blocks in vBS1 and vBS2 every $T_1$ slots during the pre-caching phase. Therefore, we know that (27) can be solved by simple quadratic programming to arrive at the optimal computing frequency $f_k^*(t)$:

$$f_k^*(t)=\left[\frac{\tau\frac{Z_k(t)}{c_k}-\tau Y_k(t)}{2Vrc_k\sum_{l=1}^n y(\Pi(d_{k,l}(t)))}\right]^+. \tag{28}$$

Therefore, we propose a step-by-step iterative ARO algorithm for optimizing the 3C resources, as summarized in Algorithm 2. Observe that the algorithm has a complexity order of $O(TK(d^{max}S+n))$ and converges with probability 1 by Theorem 4.4 of [11]. In APPENDIX A, we prove that the set of $\{\boldsymbol{X}^*(t),\boldsymbol{P}^*(t),\boldsymbol{F}^*(t)\}$ is the optimal solution of (12).

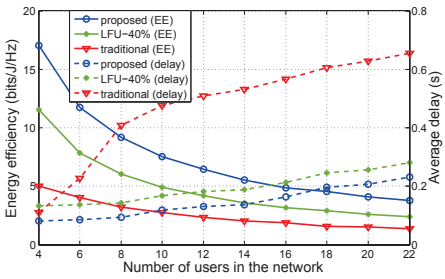


Figure 3: EE and delay versus the number of users $K$.

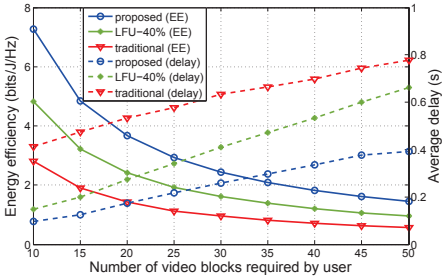


Figure 4: EE and delay versus the number $n$ of video blocks requested.

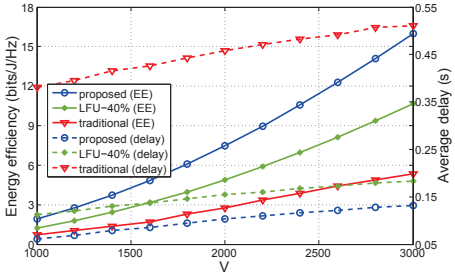


Figure 5: EE and delay versus the tradeoff factor $V$.

## IV. Numerical Results and Discussions

In this section, we compare the performance of our 3C scheme to that of the traditional caching scheme, where the most popular videos are cached at the edge without compression and the remaining videos are transmitted over the backhaul link. This scheme also adopts the NAEC decision to evict and replace video blocks in vBS1. For simplicity, the subcarrier bandwidth $B$ is normalized to 1. The parameters are given as follows: $V = 2000$, $K = 10$, $S = 64$, $L = 100$, $M = 1$ kbits, $C_n = 30$, $n = 10$, $c_k = 2000$ cycle/bit, $r = 10^{-27}$ W·s$^3$/cycle$^3$, $f_k^{max} = 10^{10}$ cycle/s, $P_k^{max} = (40/K)$ W, $P_k^c = 0.2$ W, $\alpha = 1$, $\tau = 0.15$ s/slot, $R_k^{av} = 1$ Mbit/s and $(\beta + \omega) = 0.4s$.

Firstly, to show the advantages of our NAEC decision in terms of eviction and replacement, we use the LFU-40% caching (only fixed update 40% of video blocks in vBS1 by the LFU principle in each update) and the static caching (video contents cached in vBS1 are always unchanged) regimes for comparison. We evaluated the cache hit rate of these cache decisions in Fig. 2. It can be seen that our proposed cache decision maintains a higher cache hit rate (more than 80%) than the other decision regimes after initialization.

Fig. 3 characterizes both the EE and delay versus the number of users $K$ for three schemes. For our proposed 3C scheme, the EE decreases near-exponentially upon increasing $K$, while the delay increases almost linearly. It is expected that the increase of the number of users has a negative impact on the network's performance. For the LFU-40% regime and for the traditional scheme, the variation of EE and delay is similar to that of our proposed scheme, but its performance is inferior. The delay of the traditional scheme is seen to be higher than that of the other two schemes, because its backhaul transmission imposes a higher delay. Therefore, if the edge storage space is sufficient, the MNO should first consider using the proposed 3C scheme, which saves more energy consumption and reduces delay than the traditional scheme and the LFU-40% regime.

Fig. 4 depicts both the EE and delay versus the number $n$ of video blocks requested by user $k$ in slot $t$ under the three schemes considered. Naturally, a higher $n$ implies a higher traffic load in the network. Observe that as the traffic loads increase, the EE of the three schemes decreases near-exponentially, which indicates that the number of video blocks also has an adverse effect on the network's performance. Additionally, in our proposed scheme, the delay is lower, whilst the EE is higher than that of the other two schemes. When $n$ exceeds 45, the delay of the proposed scheme saturates within about 0.4s due to the limits of maximum queue lengths controlled by $\beta$ and $\omega$.

Fig. 5 focuses on the EE and delay versus the tradeoff factor $V$ of the three schemes. It can be seen that both the EE and delay increase with the increase of $V$ for all the schemes. As observed, the higher the delay, the higher the EE. Explicitly, they cannot be improved at the same time, indicating their tradeoff. Therefore, $V$ serves as a control parameter facilitating the improvement of EE at the expense of the delay upon increasing $V$. Observe that the EE of the proposed scheme is always better than that of the other schemes.

## V. Conclusions

In this paper, we designed a 3C scheme for optimizing the EE and delay of edge caching-based video streaming by NFV. Then, we proposed the NAEC decision for pre-caching and designed the ARO algorithm for optimizing the computing, communication and caching resources with the aid of Lyapunov optimization. Finally, the numerical results demonstrated that our proposed scheme has better performance and efficiently reduces the users' delay. Additionally, the associated tradeoff between EE and delay may be readily controlled by the factor $V$.

## Appendix A
## Proof of Optimal Solution

As presented in Section 3.2 of [13], we can view $\sum_{k=1}^{K} R_k(t)$ in (19) as a perspective function of the concave function $\log_2\left[1 + \frac{p_{k,s}(t)|h_{k,s}(t)|^2}{(\frac{B}{S})N_0}\right]$, which is jointly concave in $\boldsymbol{X}(t)$ and $\boldsymbol{P}(t)$. Furthermore, $\sum_{k=1}^{K} P_k(t)$ in (19) is a linear function of $\boldsymbol{X}(t)$ and $\boldsymbol{P}(t)$, respectively, and the constraints in (19) are all linear constraints. Therefore, the problem (19) is convex. By exploiting the property of convex problems that the local minimum is the global minimum, the optimal solution $\{\boldsymbol{X}^*(t), \boldsymbol{P}^*(t)\}$ obtained by the classic Lagrange multiplier method is the global optimal solution of (13).

Similarly, (27) is a convex problem, because its objective is a single-variable quadratic function and its constraint is linear. Therefore, the optimal solution of $\boldsymbol{F}^*(t)$ is also the globally optimal solution of (24) according to the properties of convex problems. Furthermore, our proposed problem of (12)

can be decomposed into two independent subproblems (13) and (24), so the set $\{\boldsymbol{X}^*(t), \boldsymbol{P}^*(t), \boldsymbol{F}^*(t)\}$ of the optimal solution obtained by the pair of subproblems is the optimal solution of (12).

## REFERENCES


[1] D. Liu, B. Chen, C. Yang and A. F. Molisch, "Caching at the wireless edge: design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22-28, Sept. 2016.

[2] F. Gabry, V. Bioglio and I. Land, "On Energy-Efficient Edge Caching in Heterogeneous Networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3288-3298, Dec. 2016.

[3] S. Zhang, N. Zhang, P. Yang and X. Shen, "Cost-Effective Cache Deployment in Mobile Heterogeneous Networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11264-11276, Dec. 2017.

[4] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen and W. Zhu, "Understanding Performance of Edge Content Caching for Mobile Video Streaming," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1076-1089, May 2017.

[5] C. Li, L. Toni, J. Zou, H. Xiong and P. Frossard, "QoE-Driven Mobile Edge Caching Placement for Adaptive Video Streaming," in *IEEE Trans. Multim.*, vol. 20, no. 4, pp. 965-984, Apr. 2018.

[6] G. Gao, Y. Wen and J. Cai, "vCache: Supporting Cost-Efficient Adaptive Bitrate Streaming," *IEEE MultiMedia*, vol. 24, no. 3, pp. 19-27, Aug. 2017.

[7] Y. Sun, M. Peng, S. Mao and S. Yan, "Hierarchical Radio Resource Allocation for Network Slicing in Fog Radio Access Networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3866-3881, Apr. 2019.

[8] C. Wang, F. R. Yu, C. Liang, Q. Chen and L. Tang, "Joint Computation Offloading and Interference Management in Wireless Cellular Networks with Mobile Edge Computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432-7445, Aug. 2017.

[9] Donghee Lee et al., "LRFU: a spectrum of policies that subsumes the least recently used and least frequently used policies," in *IEEE Transactions on Computers*, vol. 50, no. 12, pp. 1352-1361, Dec. 2001.

[10] C. Liu, M. Bennis and H. V. Poor, "Latency and Reliability-Aware Task Offloading and Resource Allocation for Mobile Edge Computing," 2017 IEEE Globecom Workshops, Singapore, 2017, pp. 1-7.

[11] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan and Claypool, 2010.

[12] D. P. Bertsekas and R. G. Gallager, *Data Networks (2nd edition)*. Prentice Hall, 1992.

[13] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.:Cambridge Univ. Press, 2004.

[14] M. J. Neely, "Dynamic optimization and learning for renewal systems," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 32 - 46, Jan. 2013.

[15] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439-1451, Aug. 2006.