

Edge-Caching Wireless Networks: Performance Analysis and Optimization

Thang X. Vu, *Member, IEEE*, Symeon Chatzinotas, *Senior Member, IEEE*, and
Bjorn Ottersten, *Fellow, IEEE*

Abstract

Edge-caching has received much attention as an efficient technique to reduce delivery latency and network congestion during peak-traffic times by bringing data closer to end users. Existing works usually design caching algorithms separately from physical layer design. In this paper, we analyse edge-caching wireless networks by taking into account the caching capability when designing the signal transmission. Particularly, we investigate multi-layer caching where both base station (BS) and users are capable of storing content data in their local cache and analyse the performance of edge-caching wireless networks under two notable uncoded and coded caching strategies. Firstly, we calculate backhaul and access throughputs of the two caching strategies for arbitrary values of cache size. The required backhaul and access throughputs are derived as a function of the BS and user cache sizes. Secondly, closed-form expressions for the system energy efficiency (EE) corresponding to the two caching methods are derived. Based on the derived formulas, the system EE is maximized via precoding vectors design and optimization while satisfying a predefined user request rate. Thirdly, two optimization problems are proposed to minimize the content delivery time for the two caching strategies. Finally, numerical results are presented to verify the effectiveness of the two caching methods.

Index terms— edge-caching, energy efficiency, beamforming, optimization.

I. INTRODUCTION

Future wireless networks will have to address stringent requirements of delivering content at high speed and low latency due to the proliferation of mobile devices and data-hungry

This research is supported, in part, by the ERC AGNOSTIC project under code R-AGR-3283 and the FNR CORE ProCAST project.

The authors are with the Interdisciplinary Centre for Security, Reliability and Trust (SnT) – University of Luxembourg, 29 Avenue John F. Kennedy, L-1855 Luxembourg. E-Mail: {thang.vu,symeon.chatzinotas, bjorn.ottersten}@uni.lu.

Parts of this work have been presented at the IEEE SPAWC 2017 [29].

applications. It is predicted that by 2020, more than 70% of network traffic will be video [1]. Although various network architectures have been proposed in order to boost the network throughput and reduce transmission latency such as cloud radio access networks (C-RANs) [2–4] and heterogeneous networks (HetNets), traffic congestion might occur during peak-traffic times. A promising solution to reduce latency and network costs of content delivery is to bring the content closer to end users via distributed storages through out the network, which is referred to content placement or caching [5]. Caching usually consists of a placement phase and a delivery phase. The former is executed during off-peak periods when the network resources are abundant. In this phase, popular content is stored in the distributed caches. The later usually occurs during peak-traffic times when the actual users’ requests are revealed. If the requested content is available in the user’s local storage, it can be served immediately without being sent via the network. In this manner, caching allows significant backhaul’s load reduction during peak-traffic times and thus mitigating network congestion [5], [6].

Most research works on caching exploit historic user requested data to optimize either placement or delivery phases [5], [8], [9]. For a fixed content delivery strategy, the placement phase is designed to maximize the local caching gain, which is proportional to the number of file parts available in the local storage. This caching method stores the contents independently and are known as *uncoded* caching. The caching gain can be further improved via multicasting a combination of the requested files during the delivery phase, which is known as *coded* caching [6], [7]. By carefully placing the files in the caches and designing the coded data, all users can recover their desired content via a multicast stream. Rate-memory tradeoff is derived in [6], which achieves a global caching gain on top of the local caching gain. This gain is inversely proportional to the total cache memory. Similar rate-memory tradeoff is investigated in device-to-device (D2D) networks [10] and secrecy constraints [11]. In [12], [13], the authors study the tradeoff between the memory at edge nodes and the transmission latency measured in normalized delivery time. The rate-memory tradeoff of multi-layer coded caching networks is studied in [14], [15]. Note that the global gain brought by the coded caching comes at a price of coordination since the data centre needs to know the number of users in order to construct the coded messages.

Recently, there have been numerous works addressing joint content caching and transmission design for cache-assisted wireless networks. The main idea is to take into account the cached content at the edge nodes when designing the link transmission to reduce the access and backhaul costs. It is shown in [16] that transmit power and fronthaul bandwidth can be reduced via cache-

aware multicast beamforming design and power allocation. The impact of wireless backhaul on the energy consumption was studied in [17]. The authors in [18] propose a joint optimization of caching, routing and channel assignment via two sub-problems called restricted master and pricing. The performance of caching wireless D2D networks are analysed in [19–22]. In [20], the authors study D2D networks which allow the storage of files at either small base stations or user terminals. Taking into account the wireless fading channels, a joint content replacement and delivering scheme is proposed to reduce the system energy consumption. The throughput-outage trade-off of the mmWave underlying D2D networks under a simplified grid topology is derived in [21]. The stochastic performance of caching wireless networks is analysed in [23], in which the nodes' locations are modelled as a Poisson point process (PPP). The average ergodic downlink rate and outage probability are studied when cache capability is present at three tiers of base station (BS), relay and D2D pairs. In [24], success delivery rate is studied in cluster-centric networks, which group small base stations (SBSs) into disjoint clusters. In this work, the SBSs within one cluster share a cache which is divided into two parts: one contains the most popular files, and one comprises different files which are most popular locally. The authors in [25] study effects of mobility on the caching wireless networks via a random-walk assumption of node mobilities. In [26], a low-complexity greedy algorithm is proposed to minimize the content delivering delay in cooperative caching C-RANs. Energy efficiency (EE) of cache-assisted networks are analysed in [27], [28]. Focusing on the content placement phase in heterogeneous networks, the authors in [27] study the trade-off between the expected backhaul rate and energy consumption. The impact of caching is analysed in [28] via close-form expression of the approximated network EE. We note that these works consider either only the uncoded caching method or the caching at higher layers separated from the signal transmission.

In this paper, we investigate the performance of edge-caching wireless networks in which multi-layer caches are available at either user or edge nodes. Our contributions are as follows:

- Firstly, we investigate the performance of edge-caching networks under two notable *uncoded* and inter-file *coded* caching strategies¹. In particular, we compute the required throughputs on the backhaul and access links for both caching strategies with arbitrary cache sizes.
- Secondly, we derive a closed-form expression for the system EE, which reveals insight contributions of cache capability at the BS and users. Based on the derived formula, we

¹The inter-file coded caching is different from intra-file coded caching method.

maximize the system EE subject to a quality-of-service (QoS) constraint taking into account the caching strategies. The maximum EE is obtained in closed-form for zero-forcing (ZF) precoding and suboptimally solved via semi-definite relaxation (SDR) design. Our paper differs from [27], [28] as following. We focus on the delivery phase, while [28] considers the placement phase. We consider multi-layer cache and the two caching strategies, while [27] only considers caching available at the BS with an uncoded caching algorithm.

- Thirdly, we analyse and minimize the delivery time for the two caching strategies via two formulated problems which jointly optimize the beamforming design and power allocation. Our method is fundamentally different from [12] which studies the latency limit from information-theoretic perspectives. Compared with [26], which studies only uncoded caching at higher layers, we consider both caching strategies jointly with the signal transmission.
- Finally, the analysed EE and delivery time are verified via selective numerical results. We show an interesting result that the uncoded-caching is more energy-efficient only for the small user cache sizes. This result is different from the common understanding that the coded caching always outperforms the uncoded caching in terms of total backhaul's throughput.

The rest of this paper is organised as follows. Section II presents the system model and the caching strategies. Section III analyses the system energy efficiency. Section IV presents the proposed EE maximization algorithms. Section V minimizes the delivery time. Section VI derives the EE for general content popularity. Section VII shows numerical results. Finally, Section VIII concludes the paper.

Notation: $(\cdot)^H$, $(x)^+$ and $\text{Tr}(\cdot)$ denote the Hermitian transpose, $\max(0, x)$ and the trace(\cdot) function, respectively. $\lfloor x \rfloor$ denotes the largest integer not exceeding x .

II. SYSTEM MODEL

We consider the downlink edge-caching wireless network in which a data centre serves K distributed users, denoted by $\mathcal{K} = \{1, \dots, K\}$, via one BS, as depicted in Figure 1. This model can also be applied in various practical scenarios in which the users can be replaced by various cache-assisted edge nodes, e.g., edge nodes in fog radio access networks (F-RAN), small-cell BSs in HetNet. The L -antenna BS, with $L \geq K$, serves all users via wireless access networks and connects to the data centre via an error-free, bandwidth-limited backhaul link. The wireless transmissions are subjected to block Rayleigh fading channels, in which the channel fading coefficients are fixed within a block and are mutually independent across the users. The block

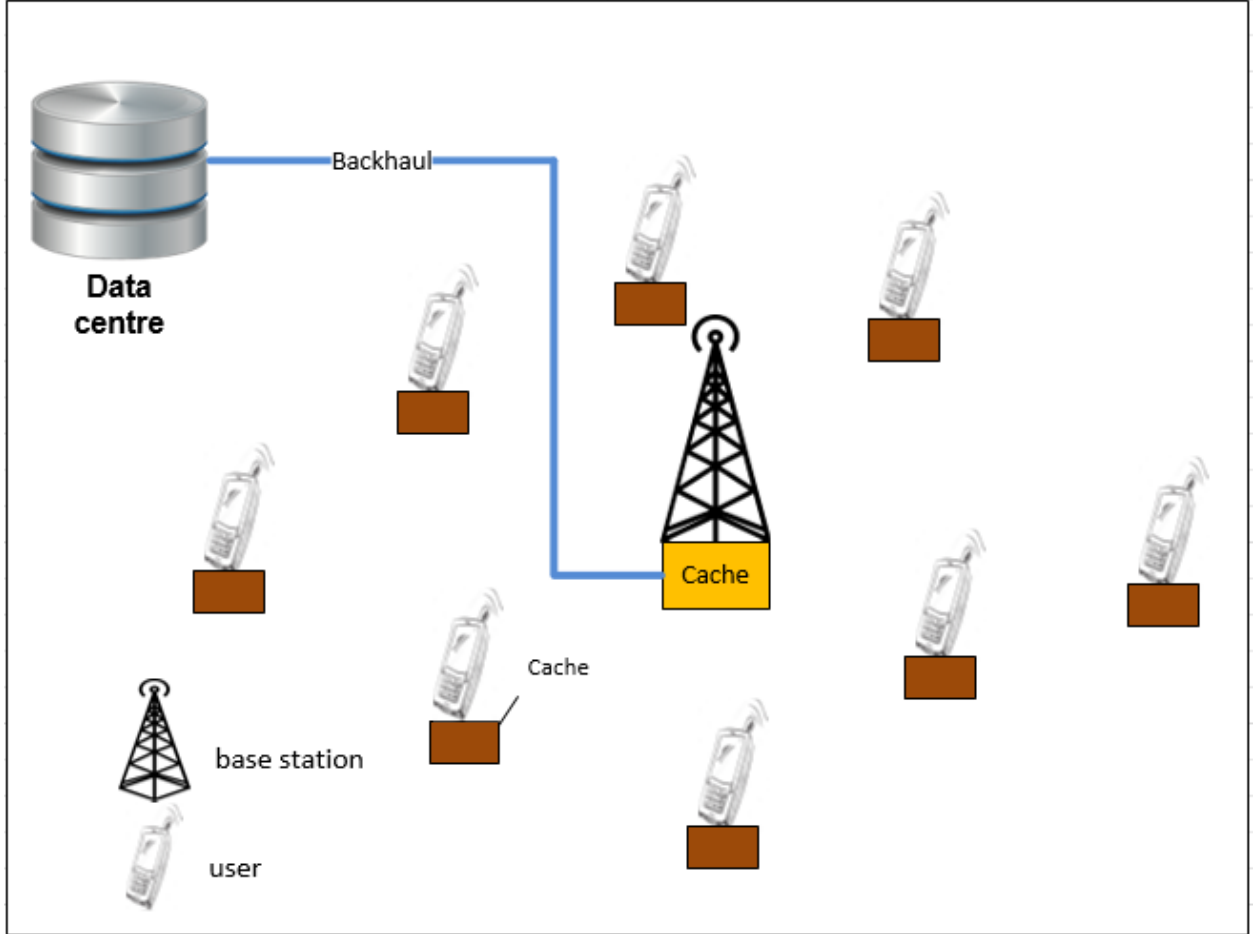


Fig. 1: Multiple-layer cache-assisted wireless networks.

duration is assumed to be long enough for the users to be served the requested files. The data centre contains N files of equal size of Q bits and is denoted by $\mathcal{F} = \{F_1, \dots, F_N\}$. In practice, unequal size files can be divided into trunks of subfiles which have the same size.

A. Caching model

We consider multiple-layer caching networks in which both the BS and users are equipped with a storage memory of size M_b and M_u files, with $0 \leq M_b, M_u \leq N$, respectively. We consider off-line caching, in which the *content placement phase* is executed during off-peak times [6]. For robustness, we consider the completely distributed placement phase in which the BS is unaware of user cache's content. In particular, the BS stores $\frac{M_b Q}{N}$ (non-overlapping) bits of every file in its cache, which are randomly chosen². Similarly, each user stores $\frac{M_u Q}{N}$ bits of

²There may exist a better cache placement at the BS at the expense of coordination.

every file in its cache under the uncoded caching strategy³. The placement phase at the user caches under the coded caching is similar to [6]. The total number of bits stored at the BS and user caches are respectively $M_b Q$ and $M_u Q$ bits, which satisfy the memory constraints.

At the beginning of the *delivery phase*, each user requests one file from the library. In order to focus on the interplay between the EE and cache capabilities, we consider the worst case in which the users tend to request different files and the content popularity follows a uniform distribution [6]. The general case of content popularities, e.g., Zipf distribution, will be studied in Section VI. Denote d_1, \dots, d_K as the file indices requested by user 1, ..., K , respectively. If the requested bits (or subfile) is in its own cache, they can be served immediately. Otherwise, this subfile is sent from the BS's cache or the data centre through the backhaul link. We consider two notable caching methods for the delivery phase: uncoded caching and coded caching.

1) *Uncoded caching*: This strategy sends parts of the requested files to each user independently. We note that the users do not know the cache content of each other. The advantage of this method is the robustness and it does not require coordination. The total number of bits transmitted through the backhaul link, $Q_{\text{unc,BH}}$, and the access link, $Q_{\text{unc,AC}}$, are given in the following proposition.

Proposition 1: Under the uncoded caching strategy, the total number of bits transmitted through the backhaul and access links are $Q_{\text{unc,BH}} = KQ \left(1 - \frac{M_u}{N}\right) \left(1 - \frac{M_b}{N}\right)$ and $Q_{\text{unc,AC}} = KQ \left(1 - \frac{M_u}{N}\right)$, respectively.

Proof: See Appendix A. ■

2) *Coded caching*: In coded caching strategy, the data centre first intelligently encodes the requested files and then sends them to the users. We note that this strategy requires the number of users in order to construct the coded messages for all users.

Proposition 2: Let $m = \lfloor \frac{KM_u}{N} \rfloor \in \mathbb{Z}^*$ and $\delta = \frac{KM_u}{N} - m$ with $0 \leq \delta < 1$. Under the coded-caching strategy, the throughput (in bits) on the access links is $Q_{\text{cod,AC}} = (1 - \delta) \frac{Q(K-m)}{m+1} + \delta \frac{Q(K-m-1)}{m+2}$, and the backhaul throughput is $Q_{\text{cod,BH}} = (1-\delta) \left(1 - \left(\frac{M_b}{N}\right)^{m+1}\right) \frac{Q(K-m)}{m+1} + \delta \left(1 - \left(\frac{M_b}{N}\right)^{m+2}\right) \frac{Q(K-m-1)}{m+2}$.

Proof: We consider two cases: i) $M_u \in \{0, \frac{N}{K}, \frac{2N}{K}, \dots, \frac{(K-1)N}{K}\}$ and ii) M_u has arbitrary value within $(0, N)$.

Case 1: $M_u \in \{0, \frac{N}{K}, \frac{2N}{K}, \dots, \frac{(K-1)N}{K}\}$

In this case, the user cache M_u is multiple times of $\frac{N}{K}$. Denote $m = \frac{M_u K}{N} \in \{0, 1, 2, \dots, K-1\}$.

³If $\frac{M_b Q}{N}$ or $\frac{M_u Q}{N}$ is not an integer, we round up this ratio to the closest integer and perform zero-padding to the last.

When $m = 0$, it is straightforward to see that $Q_{\text{cod,AC}} = QK$ and $Q_{\text{cod,BH}} = (1 - M_b/N)KQ$ since there is no cache at the users. The computation for $m \in \{1, \dots, K-1\}$ is as follows.

Computation of $Q_{\text{cod,AC}}$: We first calculate the total bits $Q_{\text{cod,AC}}$ need to be sent over the access links under the coded-caching strategy. Let $\mathcal{CC}(\mathcal{F}, \mathcal{K}, m)$ denote the coded-caching algorithm that the BS employs to serve K users. Each user is equipped with a cache of size $\frac{mN}{K}$, $m \in \mathbb{Z}^*$, and requests one file from the library \mathcal{F} . $\mathcal{CC}(\mathcal{F}, \mathcal{K}, m)$ comprises of two phases: a placement phase and a delivery phase. Due to space limitation, the details of $\mathcal{CC}(\mathcal{F}, \mathcal{K}, m)$ are omitted here but can be found in [6, Sec.V]. We only present the essential information of $\mathcal{CC}(\mathcal{F}, \mathcal{K}, m)$ which will be used in the next subsection. Each file $F_f \in \mathcal{F}$ is divided into C_K^m non-overlapped subfiles. Then each file can be expressed as $F_f = (F_{f,\mathcal{T}} | \mathcal{T} \subset \mathcal{K}, |\mathcal{T}| = m)$, where \mathcal{T} is any subset of \mathcal{K} consisting of m different elements. During the delivery phase, the BS multicasts $X_{\mathcal{S}} = \bigoplus_{s \in \mathcal{S}} F_{f_s, \mathcal{S} \setminus \{s\}}$ to all users, where $\mathcal{S} \subset \mathcal{K}$ with $|\mathcal{S}| = m+1$ and \oplus denotes the XOR operation. It has been shown in [6] that

$$Q_{\text{cod,AC}} = C_K^{m+1} \frac{Q}{C_K^m} = Q \frac{K-m}{1+m} \text{ (bits)}.$$

Computation of $Q_{\text{cod,BH}}$: Since the BS randomly stores parts of every file in its cache, the probability that a bit in file $F_f \in \mathcal{F}$ is prefetched at the BS cache is $p = \frac{M_b}{N}$. Now consider the transmission of signal $X_{\mathcal{S}}$. Each bit in $X_{\mathcal{S}}$ is the XORed of $m+1$ bits from $m+1$ different files. If these bits are available at the BS cache, there is no need to send this XORed bit through the backhaul. Otherwise, the data centre sends this XORed bit through the backhaul to the BS. Because these $m+1$ files are independent, the probability that this XORed bit is not sent through the backhaul is p^{m+1} . In other words, the probability that a XORed bit in $X_{\mathcal{S}}$ is sent through the backhaul is $1 - p^{m+1}$. Since there are $Q_{\text{cod,AC}}$ XORed bits, the total bits sent through the backhaul is

$$\begin{aligned} Q_{\text{cod,BH}} &= (1 - p^{m+1}) Q_{\text{cod,AC}} \\ &= \left(1 - \left(\frac{M_b}{N}\right)^{m+1}\right) \frac{Q(K-m)}{m+1}. \end{aligned}$$

Case 2: $0 < M_u < N$

This subsection calculates the throughput on the backhaul and access links for arbitrary values of the BS and user cache size. Let $m \in \mathbb{Z}^*$ and $0 < \delta < 1$ such as $M_u = (m + \delta) \frac{N}{K}$. For every file $F_f \in \mathcal{F}$, we divide it into two parts: F_f^1 consisting of the first $(1 - \delta)Q$ bits and F_f^2

consisting of the remaining δQ bits. Then the original library \mathcal{F} is decomposed into two disjoint sub-libraries $\mathcal{F}_1 = \{F_1^1, F_2^1, \dots, F_N^1\}$ and $\mathcal{F}_2 = \{F_1^2, F_2^2, \dots, F_N^2\}$. Note that the file size in \mathcal{F}_1 and \mathcal{F}_2 is $(1 - \delta)Q$ and δQ bits, respectively.

Cache placement phase: The placement phase in this case comprises of two steps. First, the data centre applies the placement phase of $\mathcal{CC}(\mathcal{F}_1, \mathcal{K}, m)$ on \mathcal{F}_1 . After this step, each user cache contains $(1 - \delta)M_u Q$ bits. Then, it applies $\mathcal{CC}(\mathcal{F}_2, \mathcal{K}, m + 1)$ on \mathcal{F}_2 . This steps results in $\delta M_u Q$ bits on each user cache. In total, each user cache is prefetched with $(1 - \delta)M_u Q + \delta M_u Q = M_u Q$ bits, which satisfies the memory constraint.

Delivery phase: We employ a time-splitting mechanism to serve the user requests. As a result, the delivery phase consists of two consecutive steps. First, the delivery phase of $\mathcal{CC}(\mathcal{F}_1, \mathcal{K}, m)$ is applied for \mathcal{F}_1 . This will costs a throughput $(1 - \delta) \frac{Q(K-m)}{m+1}$ bits. Then the delivery phase of $\mathcal{CC}(\mathcal{F}_2, \mathcal{K}, m + 1)$ is applied for \mathcal{F}_2 , which results in additional $\delta \frac{Q(K-m-1)}{m+2}$ bits⁴. Therefore, the total throughput on the access links is

$$Q_{\text{cod,AC}} = (1 - \delta)Q \frac{K - m}{m + 1} + \delta Q \frac{K - m - 1}{m + 2}.$$

We observe that the probability that each XORed bit in \mathcal{F}_1 and \mathcal{F}_2 is stored at the BS cache is q^{m+1} and q^{m+2} , respectively. Therefore, the backhaul throughput in this case is

$$\begin{aligned} Q_{\text{cod,BH}} &= (1 - \delta) (1 - q^{m+1}) Q \frac{K - m}{m + 1} \\ &\quad + \delta (1 - q^{m+2}) Q \frac{K - m - 1}{m + 2}. \end{aligned}$$

■

Proposition 2 derives the aggregated throughput on the access links under the coded-caching strategy for arbitrary values $M_u \in [0, N]$. When $\delta = 0$ and $\frac{KM_u}{N} \in \mathbb{Z}^*$, the result is shorten as $\frac{KQ(1-M_u/N)}{1+KM_u/N}$, which can also be found in [6]. Note that [6] derives the access link's throughput only for limited values of M_u such as $\frac{KM_u}{N} \in \mathbb{Z}$. In other words, Proposition 2 generalizes the result in [6] for arbitrary values of the user cache size.

B. Transmission model

This subsection describes the transmission of the requested files from the BS to users. Let $\mathbf{h}_k \in \mathbb{C}^{L \times 1}$ denote the channel vector from the BS antennas to user k , which follows a circular-

⁴This time-splitting mechanism can be seen as an implementation scheme to achieve the memory-sharing performance in [6].

symmetric complex Gaussian distribution $\mathbf{h}_k \sim \mathcal{CN}(\mathbf{0}, \sigma_{h_k}^2 \mathbf{I}_K)$, where $\sigma_{h_k}^2$ is the parameter accounting for the path loss from the BS antennas to user k . Perfect channel state information (CSI) is assumed to be available at the BS. In practice, robust channel estimation can be achieved through the transmission of pilot sequences. When a user requests a file, it first checks its own cache. If the requested file is available in its cache, it can be served immediately. Otherwise, the user sends the requested file's index to the data centre. If the requested file is not at the BS cache, it will be sent from the data centre via the backhaul. Then the BS transmits the requested file to the user via the access links.

1) *Signal transmission for uncoded caching strategy*: The data stream for each user under the uncoded caching method is transmitted independently. Denote F_{d_1}, \dots, F_{d_K} as the requested files from user $1, \dots, K$, respectively, and $\bar{F}_{d_1}, \dots, \bar{F}_{d_K}$ as parts of the requested files which are not at the user cache. First, the BS modulates \bar{F}_{d_k} in to corresponding modulated signal x_k and then sends the precoded signal through the access channels. Denote $\mathbf{w}_k \in \mathbb{C}^{L \times 1}$ as the precoding vector for user k . The received signal at user k is given as $y_k = \mathbf{h}_k^H \mathbf{w}_k x_k + \sum_{l \neq k} \mathbf{h}_k^H \mathbf{w}_l x_l + n_k$, where n_k is Gaussian noise with zero mean and variance σ^2 . The first term in y_k is the desired signal, and the second term is the inter-user interference. The signal-to-interference-plus-noise ratio at user k is $\text{SINR}_k = \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{l \neq k} |\mathbf{h}_k^H \mathbf{w}_l|^2 + \sigma^2}$. The information achievable rate of user k is

$$R_{\text{unc},k} = B \log_2 (1 + \text{SINR}_k), 1 \leq k \leq K, \quad (1)$$

where B is the access links' bandwidth.

The transmit power on the access links under the uncoded caching policy is $P_{\text{unc}} = \sum_{k=1}^K \|\mathbf{w}_k\|^2$.

2) *Signal transmission for coded caching strategy*: Obviously, one can use the transmission design derived for the uncoded caching to delivery the requested files in the coded-caching method. However, since the coded caching strategy transmits a coded message to a group of users all users during the delivery phase, using the orthogonal beams might result in resources inefficiency. Thus, we employ physical-layer multicasting [30] to precode the data for the coded caching strategy.

In the coded caching strategy, the BS will send C_K^{m+1} coded messages (of length $\frac{Q}{C_K^m}$ bits) in total to the users, each of which is received by a subset of $m+1$ users [6]. Denote by $\mathcal{S} \subset \mathcal{K}$ an arbitrary subset consisting of $m+1$ users, and by $\mathcal{S} = \{\mathcal{S} \mid |\mathcal{S}| = m+1\}$ all subsets of

$m + 1$ users. Obviously, $|\mathcal{S}| = C_K^{m+1}$. For convenience, we denote $X_{\mathcal{S}}$ as the coded message targeted for the users in \mathcal{S} . The received signal at user $k \in \mathcal{S}$ is given as $y_k = \mathbf{h}_k^H \mathbf{w}_{\mathcal{S}} x_{\mathcal{S}} + n_k$, where $\mathbf{w}_{\mathcal{S}}$ is the beamforming vector for the users in \mathcal{S} and $x_{\mathcal{S}}$ is the modulated signal of $X_{\mathcal{S}}$. The achievable rate for the users under physical-layer multicasting is

$$R_{\text{cod},\mathcal{S}} = \min_{k \in \mathcal{S}} \left\{ B \log_2 \left(1 + \frac{|\mathbf{h}_k^H \mathbf{w}_{\mathcal{S}}|^2}{\sigma^2} \right) \right\}. \quad (2)$$

The transmit power on the access links under the coded caching policy is $P_{\text{cod}} = \|\mathbf{w}_{\mathcal{S}}\|^2$.

III. ENERGY-EFFICIENCY ANALYSIS

This section analyses the EE performance of the two caching strategies.

Definition 1 (Energy efficiency): The EE measured in bit/Joule is defined as:

$$\text{EE} = \frac{KQ}{E_{\Sigma}},$$

where KQ is the total requested bits from the K users and E_{Σ} is the total energy consumption for delivering these bits.

Since the cache placement phase in off-line caching occurs much less frequently (daily or weekly) than the delivery phase, we assume the energy consumption in the placement phase is negligible and thus E_{Σ} is the energy cost in the delivery phase [6], [16].

A. EE analysis for uncoded caching strategy

The total energy cost under the uncoded caching policy is given as $E_{\text{unc},\Sigma} = E_{\text{unc},\text{BH}} + E_{\text{unc},\text{AC}}$, where $E_{\text{unc},\text{BH}}$ and $E_{\text{unc},\text{AC}}$ are the energy cost on the backhaul and access links, respectively⁵. To compute the energy consumption on the access links, we note that each user requests $\frac{Q_{\text{unc},\text{AC}}}{K}$ bits. The uncoded caching strategy sends these bits to each user independently via unicasting. Since user k requests a file at rate $R_{\text{unc},k}$, it takes $\frac{Q_{\text{unc},\text{AC}}}{K R_{\text{unc},k}}$ seconds to complete the transmission. Therefore, the total energy consumed on the access links is calculated as

$$E_{\text{unc},\text{AC}} = \frac{Q_{\text{unc},\text{AC}}}{K R_{\text{unc},k}} P_{\text{unc}} = Q \left(1 - \frac{M_u}{N} \right) \sum_{k=1}^K \frac{\|\mathbf{w}_k\|^2}{R_{\text{unc},k}}.$$

⁵In practice, E_{Σ} also includes a static energy consumption factor.

Sine the backhaul link provides enough capacity to serve the access network, the energy cost on the backhaul is modelled as

$$E_{\text{unc,BH}} = \eta Q_{\text{unc,BH}} = \eta K Q \left(1 - \frac{M_u}{N}\right) \left(1 - \frac{M_b}{N}\right),$$

where η is a constant. In practices, η can be seen as the pricing factor used to trade energy for transferred bits [16]. The actual value of η depends on the backhaul technology.

Therefore, the EE under the uncoded caching strategy is given as

$$\text{EE}_{\text{unc}} = \frac{K}{\left(1 - \frac{M_u}{N}\right) \left(\eta K \left(1 - \frac{M_b}{N}\right) + \sum_{k=1}^K \frac{\|\mathbf{w}_k\|^2}{R_{\text{unc},k}}\right)}. \quad (3)$$

It is observed from (3) that EE_{unc} is jointly determined by the cache capacities M_u and M_b and the transmitted power on the access links.

B. EE analysis for coded caching strategy

The energy cost on the backhaul link under the coded caching policy is given as $E_{\text{cod,BH}} = \eta Q_{\text{cod,BH}}$, where η is the pricing factor. In order to calculate the energy consumption on the access links, $E_{\text{cod,AC}}$, we note that the BS multicasts the coded information X_S to the users in \mathcal{S} . With the rate $R_{\text{cod},S}$, it takes $\frac{Q_{\text{cod,AC}}}{C_K^{m+1} R_{\text{cod},S}}$ seconds to send X_S . The total energy consumed by the BS in this case is $E_{\text{cod,AC}} = \frac{Q_{\text{cod,AC}}}{C_K^{m+1}} \sum_{S \in \mathcal{S}} \frac{P_{\text{cod},S}}{R_{\text{cod},S}}$. Therefore, the EE under the coded caching strategy is given as

$$\text{EE}_{\text{cod}} = \frac{KQ}{E_{\text{cod},\Sigma}} = \frac{KQ}{\eta Q_{\text{cod,BH}} + \frac{Q_{\text{cod,AC}}}{C_K^{m+1}} \sum_{S \in \mathcal{S}} \frac{P_{\text{cod},S}}{R_{\text{cod},S}}}. \quad (4)$$

From Proposition 2 we obtain

$$\text{EE}_{\text{cod}} = \frac{1 + \frac{KM_u}{N}}{\left(1 - \frac{M_u}{N}\right) \left(\eta \left(1 - \left(\frac{M_b}{N}\right)^{\frac{KM_u}{N} + 1}\right) + \frac{1}{C_K^{m+1}} \sum_{S \in \mathcal{S}} \frac{\|\mathbf{w}_S\|^2}{R_{\text{cod},S}}\right)}.$$

Similarly, the EE under the coded-caching is determined by the BS and user storage capacity and the transmit power on the access links.

C. Comparison between the two strategies

In general, the comparison between the two caching methods is complicated due to the contributions of many system parameters. In some cases, e.g., $\frac{KM_u}{N} \in \mathbb{Z}^*$, however, it is possible to explicitly reveal each method's performance. Assuming that all users are served at the same rate, e.g., $R_{\text{unc},k} = R_{\text{cod},S} = \gamma, \forall k, S$.

1) *Free-cost backhaul link*: This occurs when the BS cache is large enough to store all the files, e.g., $M_b = N$ or $\eta = 0$. All the requested files are available at either user cache or BS cache. Consequently, we have:

$$\text{EE}_{\text{unc}} = \frac{K}{\left(1 - \frac{M_u}{N}\right) \frac{P_{\text{unc}}}{\gamma}}, \quad \text{EE}_{\text{cod}} = \frac{1 + \frac{KM_u}{N}}{\left(1 - \frac{M_u}{N}\right) \frac{P_{\text{cod}}}{\gamma}}.$$

When the two methods use the same transmit power on the access links, i.e., $P_{\text{unc}} = P_{\text{cod}}$, we have $\text{EE}_{\text{unc}} > \text{EE}_{\text{cod}}$. In general, the coded caching strategy will achieve a higher EE than the uncoded caching method when $M_u > \left(\frac{P_{\text{cod}}}{P_{\text{unc}}} - \frac{1}{K}\right) N$.

2) $M_b = 0$: In this case, all the requested files which are not at the user cache will be sent from the data centre, and thus

$$\text{EE}_{\text{unc}} = \frac{1}{\left(1 - \frac{M_u}{N}\right) \left(\eta + \frac{P_{\text{unc}}}{\gamma K}\right)}, \quad \text{EE}_{\text{cod}} = \frac{1 + \frac{KM_u}{N}}{\left(1 - \frac{M_u}{N}\right) \left(\eta + \frac{P_{\text{cod}}}{\gamma}\right)}.$$

It is observed that the coded-caching strategy achieves higher EE than the uncoded caching method for the same transmit power since $\frac{KM_u}{N} > 0$ and $\frac{P_{\text{unc}}}{K} < P_{\text{cod}}$.

IV. ENERGY-EFFICIENCY MAXIMIZATION IN EDGE-CACHING WIRELESS NETWORKS

We aim at maximizing the EE in edge-caching wireless networks under the two caching strategies. The general optimization problem is formulated as follows:

$$\begin{aligned} & \text{Maximize} \quad \text{EE} \\ & \text{s.t.} \quad \text{QoS constraint,} \end{aligned} \quad (5)$$

$\{\mathbf{w}_k\}_{k=1}^K, \mathbf{w}$

where $\text{EE} \in \{\text{EE}_{\text{unc}}, \text{EE}_{\text{cod}}\}$ and $R_k \in \{R_{\text{unc},k}, R_{\text{cod}}\}$ which are given in Section II-B.

A. EE maximization for uncoded caching strategy

Let γ_k denote the QoS requirement of user k (bits per second). Without caching, it takes $t_k = \frac{Q}{\gamma_k}$ seconds to send user k the requested file. However, since parts of the requested files are available in the user cache, the BS needs to send only $(1 - \frac{M_u}{N})Q$ bits to user k . Therefore, the rate requirement taking into account the user cache is $\bar{\gamma}_k = (1 - \frac{M_u}{N})Q/t_k = (1 - \frac{M_u}{N})\gamma_k$. It is observed from (3) that for a given network topology, the BS and user cache memories are

fixed. Therefore, maximizing the EE is equivalent to minimizing the transmit power. Therefore, the problem (5) is equivalent to the following problem:

$$\underset{\{\mathbf{w}_k \in \mathbb{C}^L\}_{k=1}^K}{\text{Minimize}} \sum_{k=1}^K \frac{\|\mathbf{w}_k\|^2}{R_{\text{unc},k}}, \quad \text{s.t.} \quad \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{l \neq k} |\mathbf{h}_k^H \mathbf{w}_l|^2 + \sigma^2} \geq \zeta_k, \forall k, \quad (6)$$

where the rate constraint is replaced by an equivalent SINR constraint $\zeta_k = 2^{\frac{\bar{\gamma}_k}{B}} - 1$.

1) *Cost minimization by Zero-Forcing precoding:* In this subsection, we maximize the EE based on the ZF design because of its low computational complexity. Since the direction of the beamforming vectors are already defined by the ZF, only transmitting power on each beam needs to be optimized. Let $p_k, 1 \leq k \leq K$, denote the transmit power dedicated for user k . The precoding vector for user k is given as $\mathbf{w}_k = \sqrt{p_k} \tilde{\mathbf{h}}_k$, where $\tilde{\mathbf{h}}_k$ is the ZF beamforming vector for user k , which is the k -th column of $\mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}$, with $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_K]^T$.

Theorem 1: Under the ZF design, the uncoded caching strategy achieves the maximal EE given as

$$\text{EE}_{\text{unc}}^{\text{ZF}} = \frac{K}{\left(1 - \frac{M_u}{N}\right) \left(\eta K \left(1 - \frac{M_b}{N}\right) + \frac{\sigma^2 \sum_{k=1}^K \zeta_k \|\tilde{\mathbf{h}}_k\|^2}{\bar{\gamma}_k} \right)}.$$

Proof: By definition, $|\mathbf{h}_l^H \mathbf{w}_k|^2 = p_k \delta_{lk}$, where δ_{ij} is the Dirac delta function. Therefore, the constraint in (6) becomes $\frac{p_k}{\sigma^2} \geq \zeta_k, \forall k$. Consequently, the cost minimization problem is formulated as follows:

$$\underset{\{p_k: p_k \geq 0\}_{k=1}^K}{\text{Minimize}} \sum_{k=1}^K \frac{a_k p_k}{\log_2(1 + a_k p_k / \sigma^2)} \quad (7)$$

$$\text{s.t.} \quad p_k \geq \zeta_k \sigma^2, \forall k,$$

where $a_k = \|\tilde{\mathbf{h}}_k\|^2$.

Consider a function $f(x) = \frac{ax}{\log_2(1+bx)}$ with $a, b \geq 0$ in \mathbb{R}^+ . The derivative of $f(x)$ is $f'(x) = \frac{a}{\log_2(1+bx)} \left(1 - \frac{bx}{\log(1+bx)(1+bx)}\right) > 0, \forall x > 0$. Therefore, the objective function of (7) is a strictly increasing function in its supports. Therefore, the optimal solution of (7) is achieved at $p_k^* = \zeta_k \sigma^2$, and the minimum transmit power is $\sigma^2 \sum_{k=1}^K \zeta_k \|\tilde{\mathbf{h}}_k\|^2$. Substituting this into EE_{unc} , we obtain the proof of Theorem 1. ■

2) *Cost minimization by Semi-Definite Relaxation:* In this subsection, we maximize the EE by design the beamforming vectors and power allocation simultaneously. It is seen that (6) is a

NP-hard problem due to its non-convex objective functions as well as the constraints. Therefore, we resort to solve a suboptimal solution of (6) by minimizing the upper bound of the objective function. Since it requires $R_{\text{unc},k} \geq \bar{\gamma}_k$ to deliver the requested content to the users successfully, we have $\frac{\|\mathbf{w}_k\|^2}{R_{\text{unc},k}} \leq \frac{\|\mathbf{w}_k\|^2}{\bar{\gamma}_k}$. Due to the difference of transmission time among the users, a user who has received the requested file may not interfere the transmission of other users. Denote $\mathcal{K}_t \triangleq \{k \mid \bar{\gamma}_k \leq \frac{Q}{t}, \forall t \in [0, \frac{Q}{\min_k(\bar{\gamma}_k)}]\}$ as the subset of active users at the time of interest. Then the resorted problem is stated as

$$\underset{\mathbf{w}_k \in \mathbb{C}^L}{\text{Minimize}} \sum_{k \in \mathcal{K}_t} \frac{\|\mathbf{w}_k\|^2}{\bar{\gamma}_k}, \quad \text{s.t.} \quad \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{k \neq l \in \mathcal{K}_t} |\mathbf{h}_k^H \mathbf{w}_l|^2 + \sigma^2} \geq \zeta_k, \forall k. \quad (8)$$

We introduce new variables $\mathbf{X}_k = \mathbf{w}_k \mathbf{w}_k^H \in \mathbb{C}^{L \times L}$ and denote $\mathbf{A}_k = \mathbf{h}_k \mathbf{h}_k^H \in \mathbb{C}^{L \times L}$. Since $|\mathbf{h}_l^H \mathbf{w}_k|^2 = \mathbf{h}_l^H \mathbf{w}_k \mathbf{w}_k^H \mathbf{h}_l = \text{Tr}(\mathbf{h}_l \mathbf{h}_l^H \mathbf{w}_k \mathbf{w}_k^H) = \text{Tr}(\mathbf{A}_l \mathbf{X}_k)$, we can reformulate problem (8) as

$$\begin{aligned} & \underset{\mathbf{X}_k \in \mathbb{C}^{L \times L}}{\text{Minimize}} \quad \sum_{k \in \mathcal{K}_t} \text{Tr}(\mathbf{X}_k) \\ & \text{s.t.} \quad \text{Tr}(\mathbf{A}_k \mathbf{X}_k) \geq \zeta_k \sum_{k \neq l \in \mathcal{K}_t} \text{Tr}(\mathbf{A}_l \mathbf{X}_k) + \zeta_k \sigma^2, \forall k, \\ & \quad \mathbf{X}_k \succeq \mathbf{0}, \text{rank}(\mathbf{X}_k) = 1, \forall k. \end{aligned} \quad (9)$$

Problem (9) is still difficult to solve because the rank-one constraint is non-convex. Fortunately, the objective function and the two first constraints are convex. Therefore, (9) can be effectively solved by the SDR which is obtained by ignoring the rank one constraint. Since the SDR of (9) is a convex optimization problem, it can be effectively solved by using, e.g., the primal-dual interior point method [32]. Gaussian randomization procedure may be used to compensate the ignorance of the rank-one constraint in the SDR solution [31]. It has been shown that SDR can achieve a performance close to the optimal solution [31]. From the solution \mathbf{X}_k^* of the SDR of (9), we obtain the precoding vector \mathbf{w}_k^* . Substituting \mathbf{w}_k^* into (3) we obtain the EE of the uncoded caching strategy under SDR design.

B. EE maximization for coded caching strategy

Given the QoS requirement γ_k , user k expects to receive the requested file in $t_k = \frac{Q}{\gamma_k}$. Since each user receives only C_{K-1}^m coded messages out of C_K^{m+1} , the active time for user k is $\frac{C_{K-1}^m}{C_K^{m+1}} t_k = \frac{(m+1)Q}{K\gamma_k}$. Therefore, the required rate for user k is $\bar{\gamma}_k = (\frac{Q * C_{K-1}^m}{C_K^m}) / (\frac{(m+1)Q}{K\gamma_k}) = \frac{K-m}{m+1} \gamma_k$,

where $\frac{Q^* C_{K-1}^m}{C_K^m}$ is the number of coded bits sent to user k . Since the cache memories (4) are constant, maximizing the EE is equivalent to minimizing $\frac{P_{\text{cod}}}{R_{\text{cod},S}}$, where $R_{\text{cod},S}$ is given in (2). The optimization problem in this case is stated as

$$\underset{\mathbf{w}_S \in \mathbb{C}^{L \times 1}}{\text{Minimize}} \quad \frac{\|\mathbf{w}_S\|^2}{R_{\text{cod},S}}, \quad \text{s.t. } R_{\text{cod},S} \geq \bar{\gamma}_k, \forall k \in \mathcal{S}. \quad (10)$$

We note that problem (10) optimizes the beamforming vector for only a subset of users in \mathcal{S} . Because $\frac{P_{\text{cod}}}{R_{\text{cod},S}}$ is not convex, we instead find a suboptimal solution of problem (10) by minimizing the upper bound of $\frac{P_{\text{cod}}}{R_{\text{cod},S}}$, i.e., $\frac{P_{\text{cod}}}{R_{\text{cod},S}} \leq \frac{P_{\text{cod}}}{\bar{\gamma}_{\min,S}}$, where $\bar{\gamma}_{\min,S} = \min_{k \in \mathcal{S}} \bar{\gamma}_k$. By introducing a new variable $\mathbf{X} = \mathbf{w}_S^H \mathbf{w}_S \in \mathbb{C}^{L \times L}$, the reformulated problem is given as

$$\underset{\mathbf{X} \in \mathbb{C}^{L \times L}}{\text{Minimize}} \quad \frac{\text{Tr}(\mathbf{X})}{\bar{\gamma}_{\min,S}}, \quad \text{s.t. } \mathbf{X} \succeq \mathbf{0}; \text{ rank}(\mathbf{X}) = 1; \quad (11)$$

$$\text{Tr}(\mathbf{A}_k \mathbf{X}) \geq \sigma^2 (2^{\frac{\bar{\gamma}_{\min,S}}{B}} - 1), \forall k \in \mathcal{S}.$$

We observe that the objective function and the constraints of problem (11) are convex, except the rank-one constraint. This suggests to solve problem (11) via SDR method by ignoring the rank-one constraint. It is noted that the solution of SDR does not always satisfy the rank-one condition. Thus, Gaussian randomization procedure might be used to obtain the approximated vector from the SDR solution [31]. From the solution \mathbf{X}^* of problem (11), we obtain the precoding vector \mathbf{w}_S^* . Substituting \mathbf{w}_S^* into (4), we obtain the EE for the coding caching strategy.

V. MINIMIZATION OF CONTENT DELIVERY TIME

In this section, we aim at minimizing the average time for delivering the requested files to all users. In general, the delivery time is comprised of two parts caused by the backhaul and access links. In practice, the backhaul capacity is usually much greater than the access capacity. Therefore, we assume negligible delivery time on the backhaul link. It is also assumed that the processing time at the BS is fixed and negligible. Therefore, the total delivery time is mainly determined by the access links.

A. Minimization of delivery time for uncoded caching strategy

We would like to remind here that the uncoded caching strategy transmits independent data streams to the users. Let t_k be a time duration for the BS to transmit all the $\frac{Q_{\text{unc},\text{AC}}}{K}$ requested

bits to user k . Since the BS serves user k with rate $R_{\text{unc},k}$, we have $t_k = \frac{Q_{\text{unc},\text{AC}}}{K R_{\text{unc},k}}$ seconds. The average delivery time in the uncoded caching strategy is given as

$$\tau_{\text{unc}} = \frac{1}{K} \sum_{k=1}^K t_k = \frac{Q(1 - \frac{M_u}{N})}{K} \sum_{k=1}^K \frac{1}{R_{\text{unc},k}}.$$

The minimization of τ_{unc} is formulated as

$$\begin{aligned} & \underset{\{\mathbf{w}_k \in \mathbb{C}^L\}_{k=1}^K}{\text{Minimize}} && \frac{Q(1 - \frac{M_u}{N})}{K} \sum_{k=1}^K \frac{1}{R_{\text{unc},k}} \\ & \text{s.t.} && R_{\text{unc},k} \geq \bar{\gamma}_k, \forall k; \quad \sum_{k=1}^K \|\mathbf{w}_k\|^2 \leq P_{\Sigma}, \end{aligned} \quad (12)$$

where the first constraint is to satisfy the QoS requirement and P_{Σ} is the total transmit power.

1) *Zero-Forcing precoding design:* Let $\tilde{\mathbf{h}}_k$ be the ZF precoding vector for user k , which is the k -th column of the ZF precoding matrix $\mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}$. The beamforming vector is parallel to the ZF precoding vector as $\mathbf{w}_k = \sqrt{p_k} \tilde{\mathbf{h}}_k$, where p_k is the power allocating to user k . Note that under the ZF precoding, $\mathbf{h}_l^H \tilde{\mathbf{h}}_k = \delta_{lk}$, we thus have $R_{\text{unc},k}^{\text{ZF}} = \log_2(1 + \frac{p_k}{\sigma^2})$. Therefore, the problem (12) is equivalent to

$$\begin{aligned} & \underset{\{p_k: p_k \geq 0\}_{k=1}^K}{\text{Minimize}} && \frac{Q(1 - \frac{M_u}{N})}{K} \sum_{k=1}^K \frac{1}{\log_2(1 + p_k/\sigma^2)} \\ & \text{s.t.} && \frac{p_k}{\sigma^2} \geq \zeta_k, \forall k; \quad \sum_k p_k \|\tilde{\mathbf{h}}_k\|^2 \leq P_{\Sigma}. \end{aligned} \quad (13)$$

Proposition 3: Given the total power P_{Σ} satisfying $P_{\Sigma} \geq \sigma^2 \sum_{k=1}^K \zeta_k \|\tilde{\mathbf{h}}_k\|^2$, the problem (13) is convex and feasible.

Proof: We will show that $P_{\Sigma} \geq \sigma^2 \sum_{k=1}^K \zeta_k \|\tilde{\mathbf{h}}_k\|^2$ is the necessary and sufficient conditions of problem (13). It is straightforward to see that the constraints of (13) are convex. We will show that the objective function is also convex. Indeed, consider the function $f(x) = 1/\log_2(1 + ax)$ in \mathbb{R}^+ with $a > 0$. The second-order derivative of $f(x)$ is given as

$$\begin{aligned} f'(x) &= -\frac{a}{\log_2(1 + ax)(1 + ax)}, \\ f''(x) &= \frac{a^2}{\log_2^2(1 + ax)(1 + ax)^2} + \frac{a^2}{\log_2(1 + ax)(1 + ax)^2}. \end{aligned}$$

It is verified that the second-order derivative is always positive, thus the objective function is

convex in its support. Consequently, this problem can effectively solved by efficient algorithms, e.g., CVX [32].

Now assuming that the problem (13) is feasible. Then there exists a solution $\{\bar{p}_k\}_{k=1}^K$ which satisfies all the constraints. From the first constraint, it is straightforward to verify that $P_\Sigma \geq \sigma^2 \sum_{k=1}^K \zeta_k \|\tilde{\mathbf{h}}_k\|^2$. ■

2) *General beamforming design:* Finding the optimal solution of the original problem (12) is challenging because of the non-convex objective function. We instead propose to solve (12) sub-optimally via minimizing the upper bound of τ_{unc} . Since

$$\tau_{\text{unc}} \leq \max\{t_1, \dots, t_K\} = \frac{Q(1 - \frac{M_u}{N})}{\min\{R_{\text{unc},1}, \dots, R_{\text{unc},K}\}},$$

and $Q(1 - \frac{M_u}{N})$ is a positive constant, the suboptimal optimization of (12) is formulated as

$$\begin{aligned} & \underset{\{\mathbf{w}_k \in \mathbb{C}^L\}_{k=1}^K}{\text{Maximize}} \quad \min\{R_{\text{unc},1}, \dots, R_{\text{unc},K}\} \\ & \text{s.t.} \quad R_{\text{unc},k} \geq \bar{\gamma}_k, \forall k; \quad \sum_k \|\mathbf{w}_k\|^2 \leq P_\Sigma. \end{aligned} \quad (14)$$

By introducing an arbitrary positive variable x and resorting to SINR constraint, the above problem is equivalent to

$$\begin{aligned} & \underset{x>0, \{\mathbf{w}_k \in \mathbb{C}^L\}_{k=1}^K}{\text{Maximize}} \quad x, \quad \text{s.t.} \quad \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{l \neq k} |\mathbf{h}_k^H \mathbf{w}_l|^2 + \sigma^2} \geq x, \forall k, \\ & \quad \quad \quad x \geq \zeta_k; \quad \sum_k \|\mathbf{w}_k\|^2 \leq P_\Sigma, \end{aligned} \quad (15)$$

We introduce new variables $\mathbf{X}_k = \mathbf{w}_k \mathbf{w}_k^H$ and remind that $\mathbf{A}_k = \mathbf{h}_k \mathbf{h}_k^H$. The problem (15) is equivalent to

$$\begin{aligned} & \underset{\{\mathbf{X}_k \in \mathbb{C}^{L \times L}\}_{k=1}^K, x}{\text{Maximize}} \quad x, \quad \text{s.t.} \quad x \geq \zeta_k; \quad \sum_k \text{Tr}(\mathbf{X}_k) \leq P_\Sigma; \\ & \quad \quad \quad \text{Tr}(\mathbf{A}_k \mathbf{X}_k) - x \sum_{l \neq k} \text{Tr}(\mathbf{A}_k \mathbf{X}_l) \geq x \sigma^2, \forall k; \\ & \quad \quad \quad \mathbf{X}_k \succeq \mathbf{0}; \quad \text{rank}(\mathbf{X}_k) = 1. \end{aligned} \quad (16)$$

It is observed that the third constraint is convex for a given x . Therefore, the SDR solution of problem (16), which is obtained by ignoring the rank one constraint, can be solved via bisection.

TABLE I: ALGORITHM TO SOLVE (16)

-
1. Initialize $A_H, A_L = \zeta$, and the accuracy ϵ .
 2. $A_M = (A_H + A_L)/2$.
 3. Given A_M , if (17) is feasible, then $A_L := A_M$.
Otherwise $A_H := A_M$.
 4. Repeat step 2 and 3 until $|A_H - A_L| \leq \epsilon$.
-

The steps to solve are given in Table I.

$$\begin{aligned}
& \text{find } \{\mathbf{X}_k \in \mathbb{C}^{L \times L}\}_{k=1}^K & (17) \\
& \text{s.t. } \text{Tr}(\mathbf{A}_k \mathbf{X}_k) - A_M \left(\sum_{l \neq k} \text{Tr}(\mathbf{A}_k \mathbf{X}_l) + \sigma^2 \right) \geq 0, \forall k \\
& \sum_k \text{Tr}(\mathbf{X}_k) \leq P_\Sigma; \quad \mathbf{X}_k \succeq \mathbf{0}, \forall k.
\end{aligned}$$

B. Minimization of delivery time for coded caching strategy

The coded caching strategy multicasts the coded message \mathbf{X}_S to the users in \mathcal{S} . Since each \mathbf{X}_S contains $\frac{Q_{\text{cod,AC}}}{C_K^{m+1}}$ bits, the delivery time under coded-caching strategy is $\tau_{\text{cod}} = \frac{Q_{\text{cod,AC}}}{C_K^{m+1}} \sum_{S \in \mathcal{S}} \frac{1}{R_{\text{cod},S}}$, where $R_{\text{cod},S}$ is given in (2). Since the transmissions of X_S are independent, the optimization problem of τ_{cod} becomes minimizing the delivery time of each X_S , as follows:

$$\begin{aligned}
& \underset{\mathbf{w}_S \in \mathbb{C}^L}{\text{Minimize}} \quad \frac{1}{R_{\text{cod},S}} & (18) \\
& \text{s.t. } R_{\text{cod},S} \geq \bar{\gamma}_{\min,S}; \quad \|\mathbf{w}_S\|^2 \leq P_\Sigma.
\end{aligned}$$

By introducing new variables $x > 0$, $\mathbf{X} = \mathbf{w}_S \mathbf{w}_S^H \in \mathbb{C}^{L \times L}$ and using the equivalent SINR constraint, the above optimization is equivalent to

$$\begin{aligned}
& \underset{x, \mathbf{X} \in \mathbb{C}^{L \times L}}{\text{Maximize}} \quad x & (19) \\
& \text{s.t. } \text{Tr}(\mathbf{A}_k \mathbf{X}) \geq x \sigma^2, \forall k \in \mathcal{S}; \quad \mathbf{X} \succeq \mathbf{0}; \\
& x \geq 2^{\bar{\gamma}_{\min,S}} - 1; \quad \text{Tr}(\mathbf{X}) \leq P_\Sigma; \quad \text{rank}(\mathbf{X}) = 1.
\end{aligned}$$

Similar to the previous subsection, we observe that the first constraint in (19) is convex for a given x . Therefore, the above optimization can be solved via bisection and SDR by removing

TABLE II: ALGORITHM TO SOLVE (19)

-
1. Initialize A_H , $A_L = 2^{\bar{\gamma}_{\min, \mathcal{S}}} - 1$, and the accuracy ϵ .
 2. $A_M = (A_H + A_L)/2$.
 3. Given A_M , if (20) is feasible, then $A_L := A_M$.
Otherwise $A_H := A_M$.
 4. Repeat step 2 and 3 until $|A_H - A_L| \leq \epsilon$.
-

the rank one constraint. The steps to solve are given in Table II.

$$\begin{aligned}
 & \text{find } \mathbf{X} \in \mathbb{C}^{L \times L} \\
 & \text{s.t. } \text{Tr}(\mathbf{A}_k \mathbf{X}) - A_M \sigma^2 \geq 0, \forall k \in \mathcal{S} \\
 & \text{Tr}(\mathbf{X}) \leq P_\Sigma; \quad \mathbf{X} \succeq \mathbf{0}.
 \end{aligned} \tag{20}$$

VI. NON-UNIFORM FILE POPULARITY DISTRIBUTION

In most practical cases, the content popularity does not follow uniform distribution. In fact, there are always some files which are more frequently requested than the others. In this section, we consider arbitrary user content popularity and the uncoded caching strategy. Let $\mathbf{p}_k = \{q_{k,1}, \dots, q_{k,N}\}$ with $\sum_{n=1}^N q_{k,n} = 1$ denote the content popularity of user k , where $q_{k,n}$ is the probability of the n -th file being requested from user k .

The global file population at the BS is computed as follows:

$$q_{G,n} = \frac{1}{K} \sum_{k=1}^K q_{k,n}. \tag{21}$$

We consider general cache memories in which the user caches' size can be different. For convenience, let M_0 (files) denote the storage memory at the BS and M_k (files) denote the storage memory at user k . In the placement phase, each user fills its cache based on the local file popularity until full. Denote $\tilde{\mathbf{q}}_k = \Pi(\mathbf{q}_k)$ and $\tilde{\mathbf{q}}_G = \Pi(\mathbf{q}_G)$ as the sorted version in decreasing order of \mathbf{q}_k and \mathbf{q}_G , respectively. Then user k stores the first $n_k = M_k$ files in $\tilde{\mathbf{q}}_k$. Similarly, the BS stores the first $n_G = M_0$ files in $\tilde{\mathbf{q}}_G$.

In the delivery phase, the users send their requested file indices to the data centre.

Proposition 4: Let $\mathbf{D} = \{d_1, \dots, d_K\}$ denote a set of file indices which are requested by the

users. The total throughput on the access links is given as

$$Q_{\text{AC}}(\mathbf{D}) = Q \sum_{k=1}^K \mathbb{I}_{n_k}(\Pi_k(d_k)) \quad (22)$$

and the backhaul's throughput is calculated as

$$Q_{\text{BH}}(\mathbf{D}) = Q \sum_{k=1}^K \mathbb{I}_{n_G}(\Pi_G(d_k)), \quad (23)$$

where $\Pi_k(d_k)$ is the new position of file d_k after sorted by $\Pi(\mathbf{q}_k)$, and $\mathbb{I}_n(i) = 1$ if $i > n$ and 0 otherwise.

The proof of Proposition 4 is straightforward followed by checking if the requested file is available at the BS or user caches.

In this caching strategy, a user stores the whole file if it is cached. Therefore, the BS will transmit only to a subset of users $\tilde{\mathcal{K}}(\mathbf{D}) = \{k \mid \Pi_k(d_k) > n_k\}$ who do not cache the requested files. In order to minimize the energy cost, the BS applies the signal transmission design as follows:

$$\begin{aligned} & \underset{\mathbf{w}_{k \in \tilde{\mathcal{K}}(\mathbf{D})} \in \mathbb{C}^L}{\text{Minimize}} && \sum_{k \in \tilde{\mathcal{K}}(\mathbf{D})} \frac{\|\mathbf{w}_k\|^2}{\tilde{R}_{\text{unc},k}}, \\ & \text{s.t.} && \tilde{R}_{\text{unc},k} \geq \gamma, \forall k \in \tilde{\mathcal{K}}(\mathbf{D}), \end{aligned} \quad (24)$$

where $\tilde{R}_{\text{unc},k} = B \log_2 \left(1 + \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{k \neq l \in \tilde{\mathcal{K}}(\mathbf{D})} |\mathbf{h}_k^H \mathbf{w}_l|^2 + \sigma^2} \right)$.

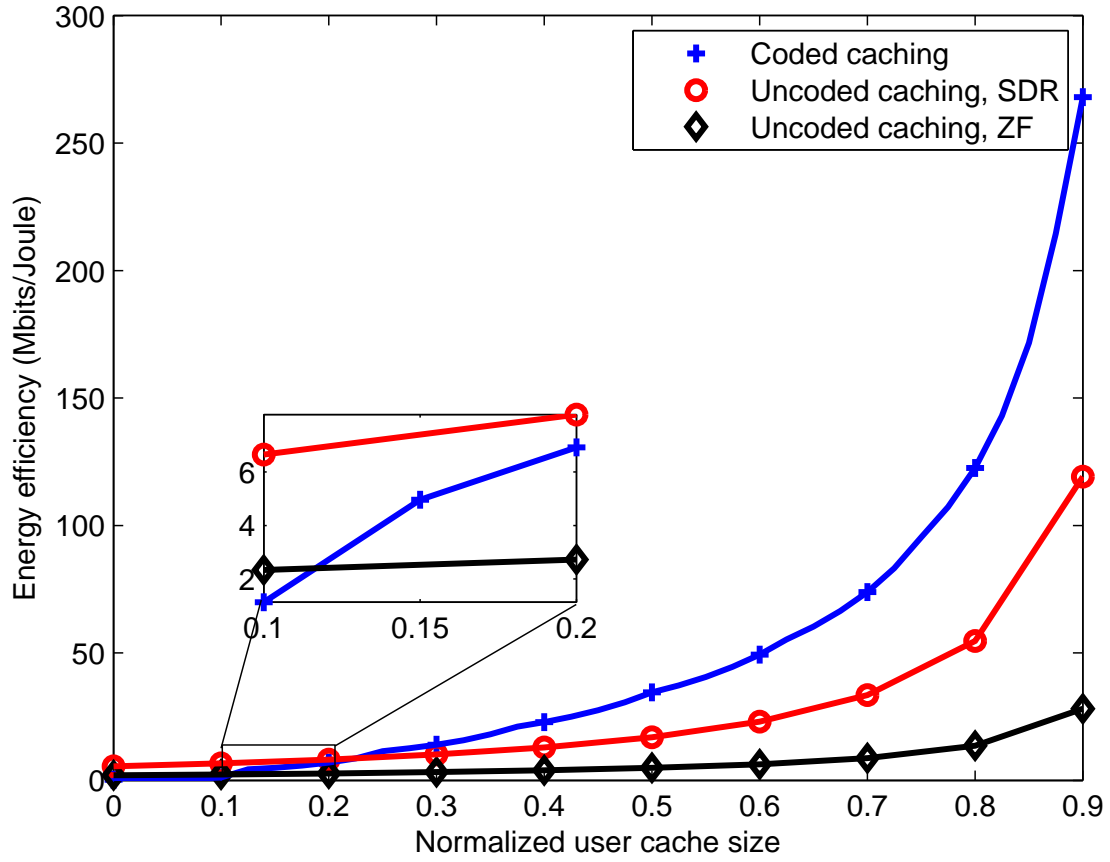
The delivery time minimization problem is formulated as:

$$\begin{aligned} & \underset{\mathbf{w}_{k \in \tilde{\mathcal{K}}(\mathbf{D})} \in \mathbb{C}^L}{\text{Minimize}} && \sum_{k \in \tilde{\mathcal{K}}(\mathbf{D})} \frac{Q}{\tilde{R}_{\text{unc},k}}, \\ & \text{s.t.} && \tilde{R}_{\text{unc},k} \geq \gamma, \forall k \in \tilde{\mathcal{K}}(\mathbf{D}). \end{aligned} \quad (25)$$

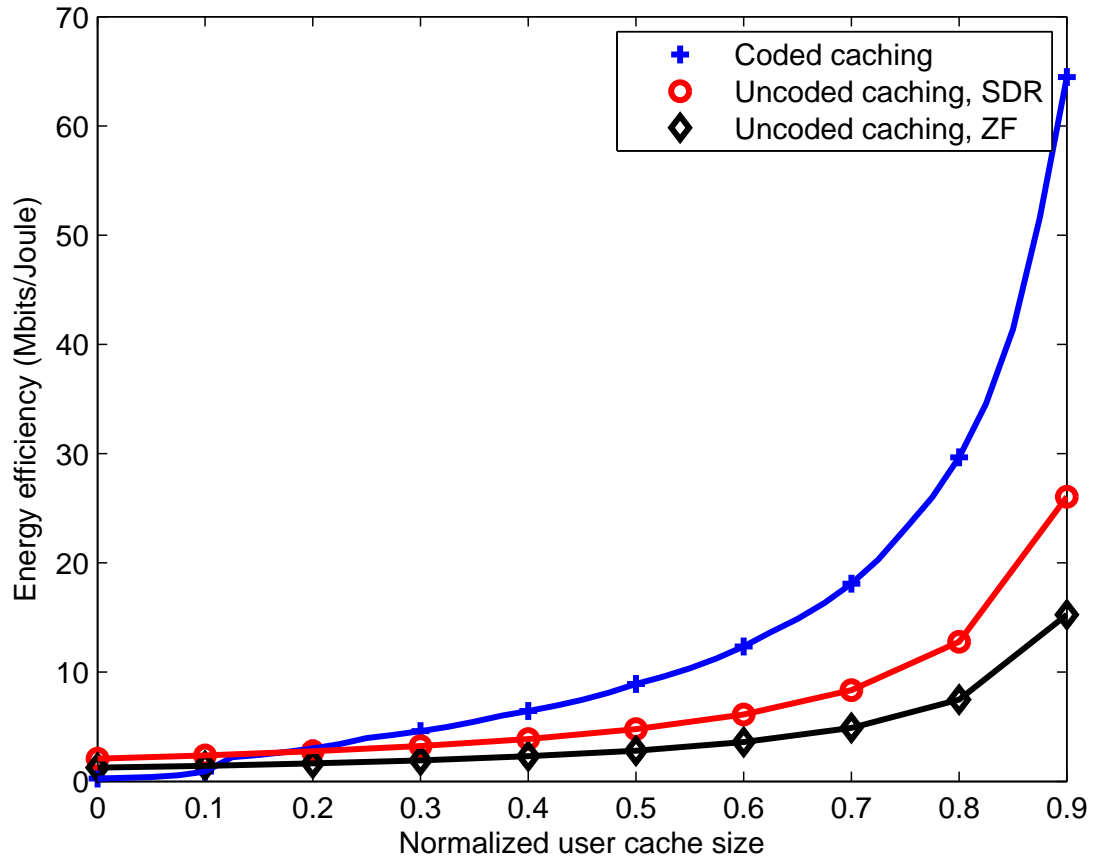
The solution of problem (24) and (25) can be found by similar techniques in Section IV-A and Section V-A, respectively.

VII. NUMERICAL RESULTS

This section presents numerical results to demonstrate the effectiveness of the studied caching policies. The results are averaged over 500 channel realizations. For ease of presentation, the uncoded caching under the general beamformer design using SDR in Section IV-A2 is named



(a) Cost-free backhaul



DRAFT

TABLE III: Simulation time in seconds, $m = K - 1$

K	Coded	Uncoded-SDR	Uncoded-ZF
4	0.197	0.384	8.7e-5
8	0.204	1.131	10e-5

as *SDR* and the Zero-forcing design in Section IV-A1 is named as *ZF* in the figures. Unless otherwise stated, the system setup is as follows: $L = 10$ antennas, $K = 8$ users, $N = 1000$ files, $B = 1$ MHz, $\eta = 10^{-6}$ bits/Joule [16], $\sigma_{h_k}^2 = 1, \forall k$, $Q = 10$ Mb, $\gamma_k = 2$ Mbps, $\forall k$.

A. Energy efficiency performance

We first study the two caching strategies when the energy consumption on the backhaul is negligible. This occurs when the BS cache is large enough to store all the files. In this case, the EE only depends on the user cache size. Figure 2a presents the EE of the two caching strategies as the function of the normalized user cache size (the user cache size M_u divided by the library size N). The EE is plotted based on the optimal precoding vectors obtained from Section IV. It is shown that the uncoded caching under the SDR design achieves higher EE than the coded caching when the normalized user cache is less than 0.2. This result suggests an important guideline for using the uncoded caching since the user cache is usually small compared to the library size in practice. When the user cache is capable of storing more than 20% of all the files, it suggests to use the coded caching for larger system EE. It is also observed that the uncoded caching under SDR design achieves higher EE than the ZF for all user cache size. This is because the SDR design is more efficient than the ZF precoding.

Figure 2b compares the EE for various user cache size when $M_b = 0.7N$. In general, the coded caching method is more efficient than the uncoded caching for most of user cache size values. Increasing user cache capability results in larger relative gain of the coded-caching compared with the uncoded method. The uncoded caching under SDR design achieves slightly better EE than the ZF design at small user cache sizes, however, at an expense of higher computational complexity as shown in Table III. From the practical point of view, ZF design is preferred in this case because of its low complexity. When M_b increases, the SDR achieves significantly higher EE than the ZF. Figure 2c presents the EE v.s. the user cache size when both BS and user cache size are small. It is shown that the uncoded caching strategy with either SDR or ZF

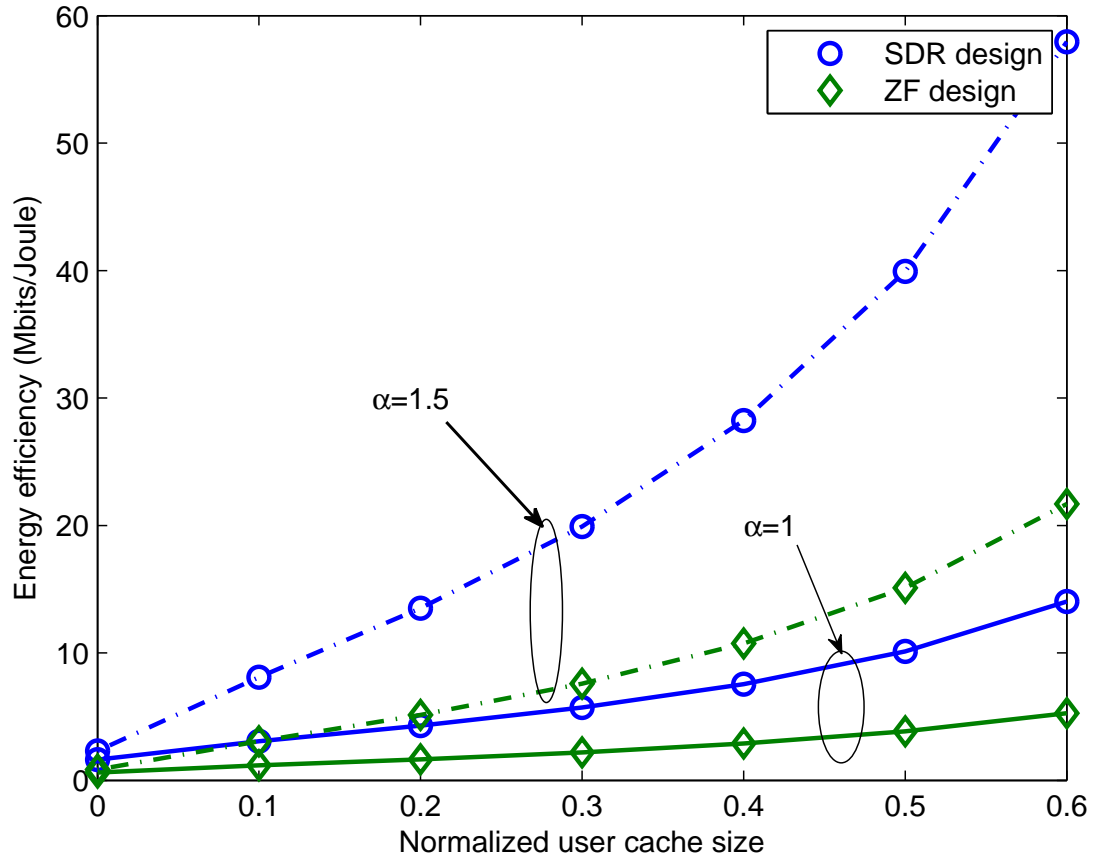
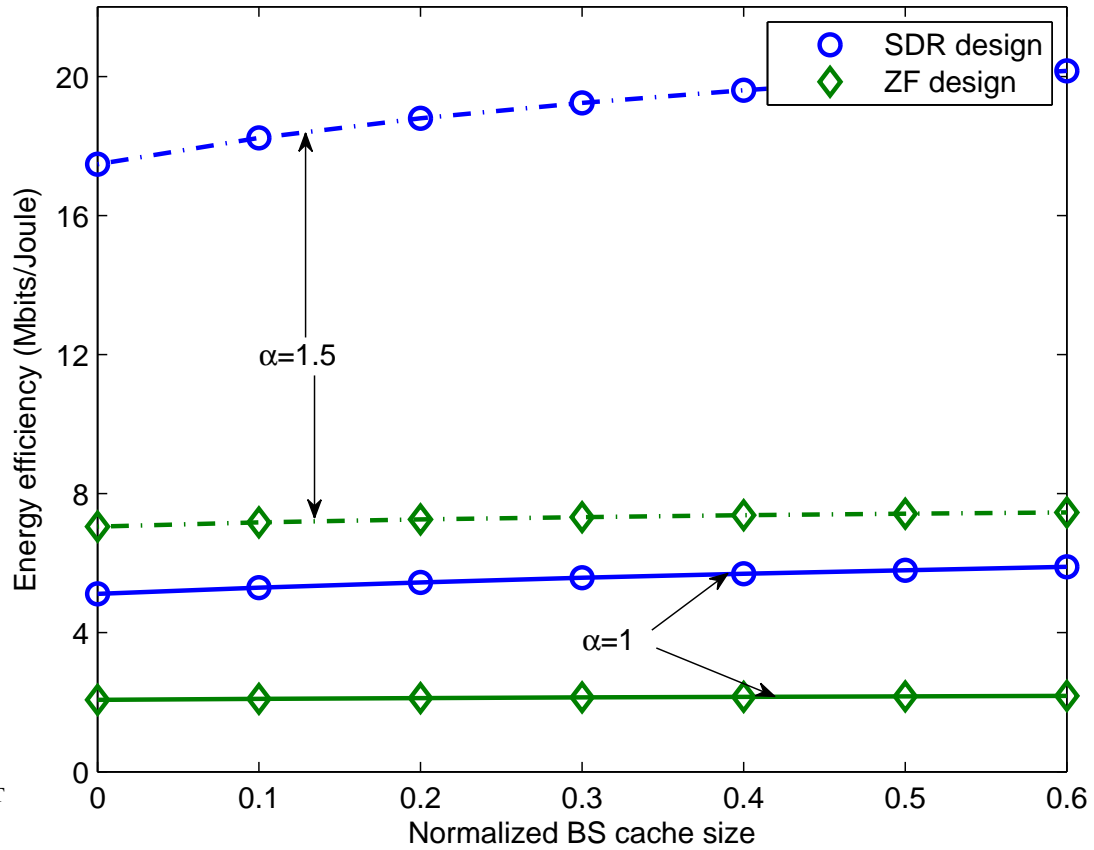
design outperforms the coded caching scheme in the observed user cache sizes, which is in line with the result in Figure 2b. Figure 2d compares the EE as a function of the BS cache size when $M_u = 0.5N$. The result shows that the caching at the BS has more impacts on both the caching strategies when the BS cache size is relatively large. It is shown that the coded-caching outperforms the uncoded caching for all values M_b . It is also shown that the SDR design achieves higher EE gain compared with the ZF as M_b increases.

Figure 3a presents the EE v.s. the normalized user cache size of the uncoded caching algorithm under Zipf content popularity distribution, i.e., $q_{k,n} = \frac{n^{-\alpha}}{\sum_{i=1}^N i^{-\alpha}}, \forall k$. It is observed that the SDR design significantly surpasses the ZF design. In particular, at 40% library size of the user cache, the SDR achieves almost 3 times EE higher than the ZF design. Greater Zipf exponent factor results in higher EE for the both designs. This is because the content distribution in this case is more centralized at some files. Figure 3b plots the EE v.s. the normalized BS cache size. Similarly, the SDR design achieves higher EE than the ZF design. Also, the BS cache size has smaller impacts on the system EE than the user cache size.

B. Delivery time performance

Figure 4 presents the delivery times of the two caching strategies as a function of the user cache size with 8 users and transmit power equal to 10 dB. It is shown that the uncoded caching strategy with both designs outperforms the coded counter part if the user cache is smaller than 30% of the library. When the cache size is larger, the coded-caching method achieves slightly smaller latency than the uncoded caching strategy. This important observation suggests the optimal caching algorithm in practical systems depending on the memory availability at the edge nodes. It is also shown that the delivery time of the uncoded caching strategy linearly depends on the cache size. This can be seen from Proposition 1 that the network throughput in the uncoded caching linearly depends on the cache size.

Figure 5 compares the delivery times of the two caching algorithms for various transmit powers. Obviously, increasing the transmit power will significantly reduce the delivery times in both strategies. When the user cache size is small (Fig. 5a), the uncoded caching strategies deliveries the requested files faster than the coded caching method, which is in line with the results in Fig. 4. When the user cache memory is capable of storing more content (Fig. 5b), the coded caching strategy is more efficient than the uncoded caching. It is also observed that

(a) $M_b = 0.3N$ 

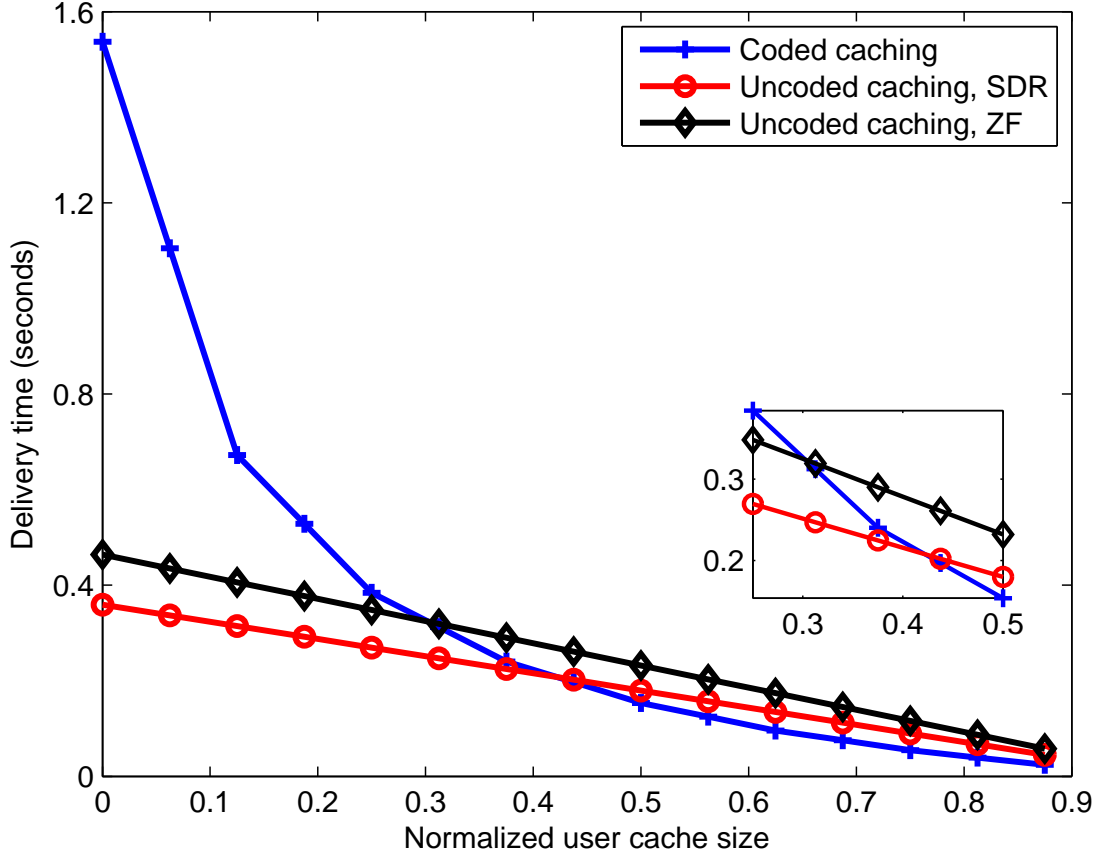


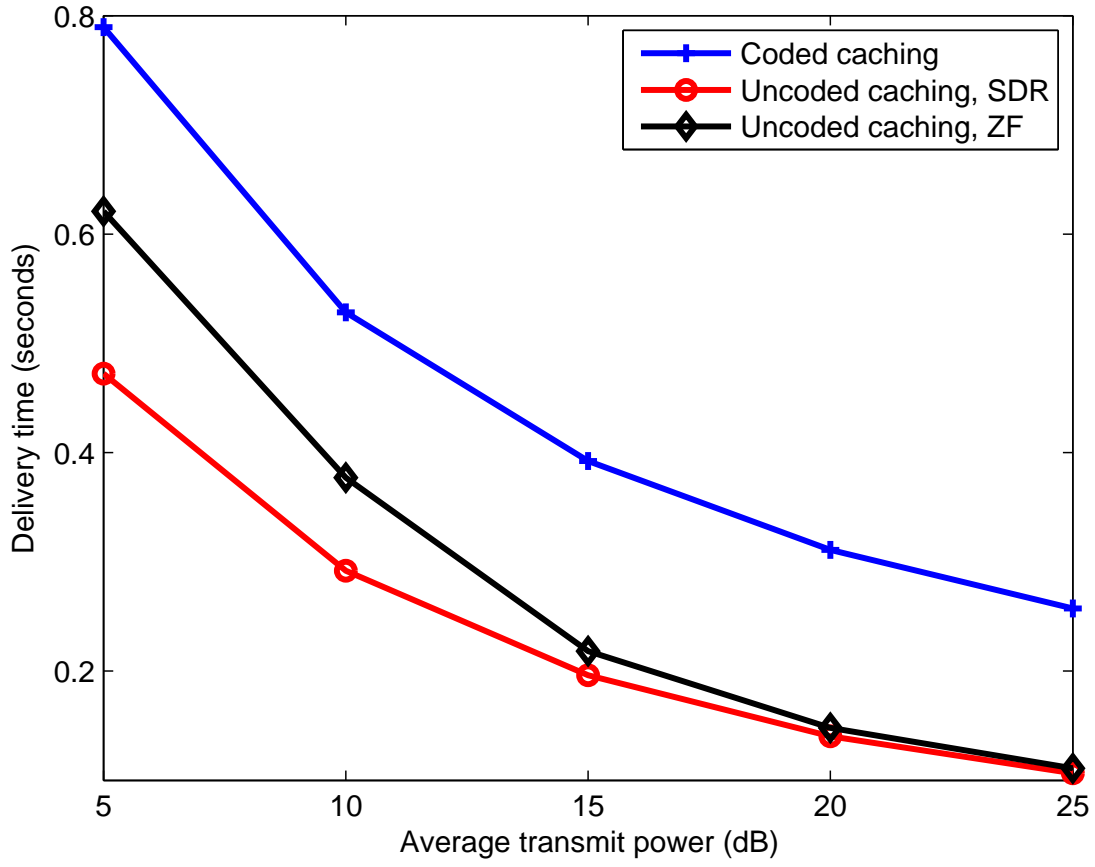
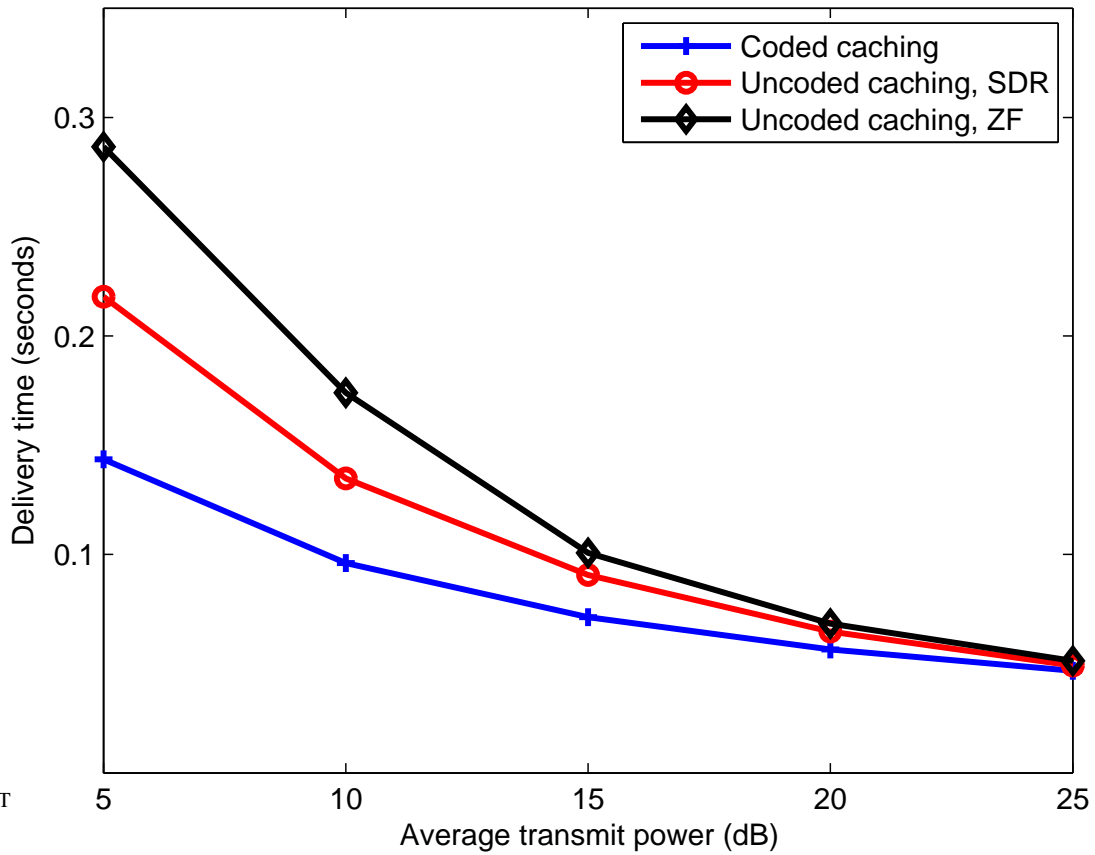
Fig. 4: Delivering time of the two caching methods v.s. the normalized user memory M_u . Average transmit power is 10 dB.

the SDR design only outperforms the ZF design for small transmit power. This is because large transmit power can supports optimal solution for both SDR and ZF designs.

Figure 6 plots the delivery times depending on the number of users K . For small K , the uncoded caching strategy slightly outperforms the coded caching method. When K increases, the coded caching tends to surpass the uncoded caching strategy. In this case, the total cache size in the network is bigger in which the coded caching algorithm is more effective.

VIII. CONCLUSIONS

We have analysed the performance of cache-assisted wireless networks under two notable uncoded and coded caching strategies. First, we have expressed the energy efficiency metric in closed-form expression for each caching strategy as a function of base station and user cache sizes and the transmit power on the access links. Based on the derived closed-form, two optimization

(a) $M_u = 0.3N$ 

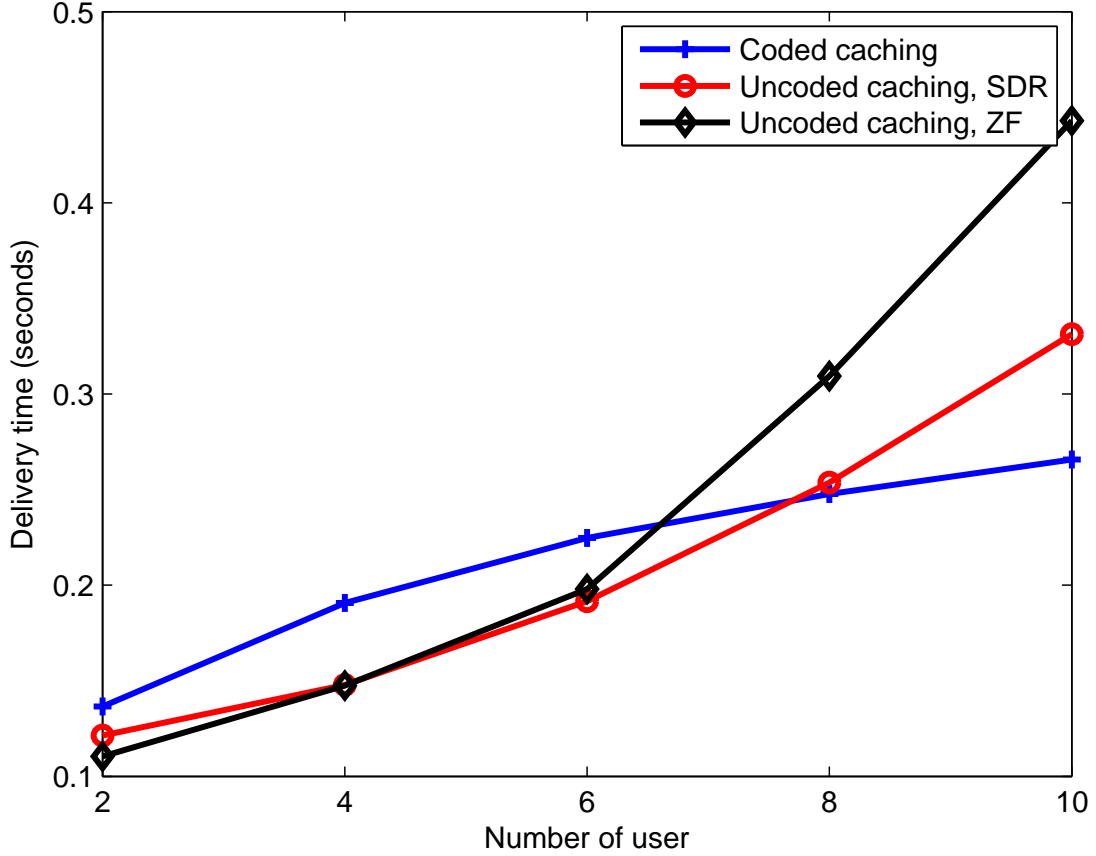


Fig. 6: Delivering time of the two caching methods v.s. the number of users. Average transmit power is 10 dB, $M_u = 0.4N$.

problems have been formulated to maximize the system EE while satisfying a predefined user rate requirement. Second, we have analysed the total delivery time for each caching strategy and designed the beamforming vectors to minimize the total delivery time. It has been shown that the uncoded caching algorithm achieves higher EE than the coded caching method only when the user cache size is small and the BS cache is large enough.

Based on the studied work, several research directions can be extended. One is to consider generic networks in which the data centre is serving multiple base stations. In this case, different backhaul constraints for each BS should be taken into account when designing the caching algorithms. Another direction is to consider the coded caching algorithm applied to non-uniform content popularity. This requires a redesign of both cache placement and delivery phases in order to take into consideration differences in user preferences.

APPENDIX A

PROOF OF PROPOSITION 1

The proof can be found by similar techniques in [7, Sec. II]. When a user requests a file, parts of the requested file are in the user cache. Since the users' requests are independent, the requested files can be either the same or different.

For any integer number $m, 1 \leq m \leq N$, there are N^m ways to choose m elements out of the set of size N , which can be further expressed as

$$N^m = \sum_{l=1}^m a_l^m \mathcal{C}_l^N,$$

where $\mathcal{C}_l^N \triangleq \frac{N!}{(N-l)!}$ and a_l^m is a constant. In the above equation, $a_l^m \mathcal{C}_l^N$ is the number of choices of m elements out of N which contains l different elements. By using the inductive method, we can obtain:

$$a_l^m = \begin{cases} 1, & \text{if } l = 1 \text{ or } m \\ ma_l^{m-1} + a_{l-1}^{m-1}, & \text{if } 1 < l < m \end{cases}$$

For a choice comprising of l different values, the BS needs to send $lQ(1 - M_u/N)$ subfiles to the users. Therefore, the average access throughput is calculated as

$$\begin{aligned} Q_{\text{unc,AC}} &= \frac{1}{N^K} \sum_{l=1}^K lQ a_l^K \mathcal{C}_l^N \left(1 - \frac{M_u}{N}\right) \\ &= \sum_{l=1}^K \frac{lQ a_l^K}{N^{K-l}} \left(1 - \frac{M_u}{N}\right) \prod_{i=1}^l \frac{N-l+i}{N}. \end{aligned} \quad (26)$$

It is observed that the library size N is usually very large compared to K , thus $\frac{N-l+i}{N} \simeq 1, \forall 1 \leq i \leq l$ and

$$\frac{l a_l^K}{N^{K-l}} \simeq \begin{cases} 0, & \text{if } l < K \\ K, & \text{if } l = K \end{cases}. \quad (27)$$

From (26) and (27) we obtain:

$$Q_{\text{unc,AC}} \simeq KQ \left(1 - \frac{M_u}{N}\right). \quad (28)$$

To compute the backhaul throughput, we note that the BS randomly cache $\frac{M_b}{N}$ parts of every file. Therefore, the probability that a bit is stored at the BS cache is $\frac{M_b}{N}$. Finally, since the BS

is the caching at the BS and users independent, we obtain $Q_{\text{unc,BH}}$ in Proposition 1.

REFERENCES

- [1] Cisco, “Cisco visual networking index: Global mobile data traffic forecast update 2016-2021,” 2017, white paper.
- [2] T. X. Vu, H. D. Nguyen, T. Q. S. Quek, and S. Sun, “Adaptive cloud radio access networks: compression and optimization,” *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 228–241, Jan. 2017.
- [3] T. X. Tran and D. Pompili, “Dynamic Radio Cooperation for User-Centric Cloud-RAN With Computing Resource Sharing,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2379–2393, Apr. 2017.
- [4] T. X. Tran, A. Hajisami, and D. Pompili, “QuaRo: A queue-aware robust coordinated transmission strategy for downlink C-RANs,” in *Proc. IEEE Int. Conf. Sensing, Commun. and Netw.*, London, 2016, pp. 1-9.
- [5] S. Borst, V. Gupta, and A. Walid, “Distributed caching algorithms for content distribution networks,” in *Proc. IEEE Int. Conf. Comput. Commun.*, Mar. 2010, pp. 1–9.
- [6] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [7] T. X. Vu, S. Chatzinotas, and B. Ottersten, “Coded caching and storage allocation in heterogeneous networks,” in *Proc. IEEE Wireless Commun. Netw. Conf.*, San Francisco, CA, 2017, pp. 1–5.
- [8] K. C. Almeroth and M. H. Ammar, “The use of multicast delivery to provide a scalable and interactive video-on-demand service,” *IEEE J. Sel. Areas Commun.*, vol. 14, no. 6, pp. 1110–1122.
- [9] D. Christopoulos, S. Chatzinotas, and B. Ottersten, “Cellular-broadcast service convergence through caching for COMP cloud RAN,” in *Proc. IEEE Symp. Commun. Veh. Tech. in the Benelux*, Luxembourg, 2015, pp. 1–6.
- [10] M. Ji, G. Caire, and A. F. Molisch, “Fundamental limits of caching in wireless D2D networks,” *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.
- [11] A. Sengupta, R. Tandon, and T. C. Clancy, “Fundamental limits of caching with secure delivery,” *IEEE Trans. Info. Forensics and Security*, vol. 10, no. 2, pp. 355–370, Feb. 2015.
- [12] A. Sengupta, R. Tandon, and O. Simeone, “Cache aided wireless networks: Tradeoffs between storage and latency,” in *Proc. Annu. Conf. Info. Sci. Syst.*, Princeton, NJ, Mar. 2016, pp. 320–325.
- [13] S. H. Park, O. Simeone, W. Lee, and S. Shamai, “Coded multicast fronthauling and edge caching for multi-connectivity transmission in fog radio access networks,” in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, Sapporo, Japan, 2017, pp. 1-5.
- [14] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, “Hierarchical coded caching,” *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212–3229, Jun. 2016.
- [15] L. Tang and A. Ramamoorthy, “Coded caching for networks with the resolvability property,” in *Proc. IEEE Int. Symp. Inf. Theory*, Barcelona, Jul. 2016, pp. 420–424.
- [16] M. Tao, E. Chen, H. Zhou, and W. Yu, “Content-centric sparse multicast beamforming for cache-enabled cloud RAN,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118–6131, Sept. 2016.
- [17] T. X. Vu, S. Chatzinotas, and B. Ottersten, “Energy Minimization for Cache-assisted Content Delivery Networks with Wireless Backhaul,” *IEEE Wireless Commun. Lett.*, vol. pp, no. pp, pp. 1–1, 2018.
- [18] A. Khreishah, J. Chakareski, and A. Gharaibeh, “Joint caching, routing, and channel assignment for collaborative small-cell cellular networks,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2275–2284, IEEE Trans. Inf. Theory. 2016.
- [19] L. Zhang, M. Xiao, G. Wu, and S. Li, “Efficient scheduling and power allocation for D2D-assisted wireless caching networks,” *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2438–2452, Jun. 2016.

- [20] M. Gregori, J. Gmez-Vilardeb, J. Matamoros, and D. Gndz, “Wireless content caching for small cell and D2D networks,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1222–1234, May 2016.
- [21] M. Ji, G. Caire, and A. F. Molisch, “Wireless device-to-device caching networks: Basic principles and system performance,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.
- [22] M. Ji, G. Caire, and A. Molisch, “The throughput-outage tradeoff of wireless one-hop caching networks,” *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6833–6859, Dec. 2015.
- [23] C. Yang, Y. Yao, Z. Chen, and B. Xia, “Analysis on cache-enabled wireless heterogeneous networks,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 131–145, Jan. 2016.
- [24] Z. Chen, J. Lee, T. Q. Quek, and M. Kountouris, “Cooperative caching and transmission design in cluster-centric small cell networks,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3401 – 3415, May 2016.
- [25] G. Alfano, M. Garetto, and E. Leonardi, “Content-centric wireless networks with limited buffers: when mobility hurts,” *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 299–311, Jan. 2016.
- [26] T. X. Tran, F. Kazemi, E. Karimi, and D. Pompili, “Mobee: Mobility-aware energy-efficient coded Caching in cloud radio access networks,” in *Proc. IEEE Int. Conf. Mobile Ad-Hoc Sensor Syst. (MASS)*, Orlando, FL, 2017, pp. 461–465.
- [27] 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.
- [28] D. Liu and C. Yang, “Energy efficiency of downlink networks with caching at base stations,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 907–922, Apr. 2016.
- [29] T. X. Vu, S. Chatzinotas, and B. Ottersten, “Energy-efficient design for edge-caching wireless networks: When is coded-caching beneficial?” in *Proc. IEEE Int. Workshop Signal Process. Wireless Commun.*, Sapporo, 2017, pp. 1–5.
- [30] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, “Transmit beamforming for physical-layer multicasting,” *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2239–2251, Jun. 2006.
- [31] Z.-Q. Luo, W. K. Ma, A. M. C. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems,” *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, Mar. 2010.
- [32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, 2004.