Coding, Multicast, and Cooperation for Cache-Enabled Heterogeneous Small Cell Networks

Jialing Liao, *Student Member, IEEE*, Kai-Kit Wong, *Fellow, IEEE*, Yangyang Zhang, Zhongbin Zheng, and Kun Yang, *Senior Member, IEEE*

***Abstract*— Caching at the wireless edge is a promising approach to dealing with massive content delivery in heterogeneous wireless networks, which have high demands on backhaul. In this paper, a typical cache-enabled small cell network under heterogeneous file and network settings is considered using maximum distance separable (MDS) codes for content restructuring. Unlike those in the literature considering online settings with the assumption of perfect user request information, we estimate the joint user requests using the file popularity information and aim to minimize the *long-term average* backhaul load for fetching content from external storage subject to the overall cache capacity constraint by optimizing the content placement in all the cells jointly. Both multicast-aware caching and cooperative caching schemes with optimal content placement are proposed. In order to combine the advantages of multicast content delivery and cooperative content sharing, a compound caching technique, which is referred to as multicast-aware cooperative caching, is then developed. For this technique, a greedy approach and a multicast-aware in-cluster cooperative approach are proposed for the small-scale networks and large-scale networks, respectively. Mathematical analysis and simulation results are presented to illustrate the advantages of MDS codes, multicast, and cooperation in terms of reducing the backhaul requirements for cache-enabled small cell networks.**

***Index Terms*— Caching, cooperation, heterogeneous networks.**

I. Introduction

THE concept of caching has recently been introduced to the physical layer for wireless content delivery networks (CDNs) to reduce peak-time traffic, latency as well as the requirement for expensive high capacity backhaul links [1]. The main idea of caching is to pre-fetch popular content at the network edge, either at the base stations (BSs) or/and user terminals (UTs) to bring the content much closer to the users. For cache enabled networks, one needs to address, e.g., where to cache, how to cache, the corresponding transmission policy, and so on.

Regarding the first question, caching can take place at the BSs or UTs. By caching at the BSs, we can reduce the traffic in backhaul and improve the energy and spectral efficiencies, while caching at the UTs adds cooperation gain and improves network scalability, facilitating device-to-device (D2D) links.

Also, cache content placement addresses what to cache. As far as content updating is concerned, caching schemes can be divided into adaptive caching and proactive caching. Adaptive caching, a.k.a. pull-based caching, works in a reactive manner by storing content in the caches on demand. In this scheme, caching decision is performed only after users have made their requests so that online algorithms, such as the least frequently used (LFU) and least recently used (LRU), can be used. As a result, the cached content in each cell is updated every time a new round of requests are made by the users. By contrast, proactive caching is a push-based approach which proactively estimates user demand patterns and performs content placement before the users make requests. Some popular schemes include common uniform placement, popularity based placement, probabilistic placement [2], partition-based placement [3] and other offline schemes. When caching contents, we can either store the entire files or fragments of the files based on file splitting to ensure diversity of the cached contents in the case that the cache capacity is relatively limited compared to the average file size. For this reason, network codes, such as maximum distance separable (MDS) codes have been utilized to construct file pieces. In optimizing content placement, the objective is usually on one of the followings: the hit ratio [2], latency [4], backhaul load [5], service cost, and so on.

Recently, considerable research has been done on physical layer caching. An information-theoretic study was first given in [5] for a homogeneous system with a single content server and several users served with a shared link. Subsequently in, e.g., [6]–[13], more complex network topologies with heterogeneous network settings have been studied for, respectively, nonuniform file popularity, file sizes and cache sizes, random requests, secure delivery, interference channel, D2D networks, and recently fog random access networks (F-RANs).

Manuscript received January 4, 2017; revised May 15, 2017 and July 13, 2017; accepted July 15, 2017. Date of publication August 4, 2017; date of current version October 9, 2017. This work was supported in part by EPSRC under Grant EP/K015893/1, in part by the Natural Science Foundation of China under Grant 61620106011, in part by the Shenzhen Key Laboratory of Artificial Microstructure Design under Grant CXB201109210099A, and in part by Shenzhen Science and Technology Plan under Grant JSGG20150917174852555. The associate editor coordinating the review of this paper and approving it for publication was J. Zhang. *(Corresponding author: Jialing Liao.)*

J. Liao and K.-K. Wong are with the Department of Electronic and Electrical Engineering, University College London, London WC1E 7JE, U.K. (e-mail: jialing.liao.14@ucl.ac.uk; kai-kit.wong@ucl.ac.uk).

Y. Zhang is with the State Key Laboratory of Meta-RF Electromagnetic Modulation Technology, Kuang-Chi Institute of Advanced Technology, Shenzhen 518057, China (e-mail: yangyang.zhang@kuang-chi.org).

Z. Zheng is with the East China Institute of Telecommunications, China Academy of Information and Communications Technology, Shanghai 200001, China (e-mail: ben@ecit.org.cn).

K. Yang is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K., and also with the School of Communication Engineering, University of Electronic Science and Technology of China, Chengdu 610051, China (e-mail: kunyang@essex.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2017.2731967

Another hot topic for cache-enabled networks is the beamforming design. In [14], instantaneous beamforming and BS activation for the cloud RAN (C-RAN) were addressed, while [15] considered the joint design of data assignment and beamforming for a cooperative multicell network both assuming a given cache content placement in a short-term time scale. In [16], beamforming and cache content placement were jointly optimized utilizing a mixed time-scale stochastic optimization scheme. In addition, performance analysis of cache-enabled wireless networks has also been extensively conducted in the literature, e.g., [17]–[20]. To summarize, those results largely analyzed cache-enabled small-cell networks using stochastic geometry to model the stochastic properties of channel fading and interference. However, the results either ignored the spatial diversity of the cached content and disabled the coordination and cooperation aspect among different cells [17], [18] or even ignored the file popularity information altogether [19], [20].

Apart from the above, emerging topics in physical layer caching also include hierarchical caching [21], mobility-aware caching [22], multicast-aware caching, cooperative caching, and caching architecture design in fog-RAN.

In this paper, our aim is to reduce the backhaul requirements by considering multicast-aware content delivery and cooperative caching as well as the service cost using MDS codes. In the following, we discuss the related work, emphasizing their many differences compared to our work.

A. Related Work

Of relevance to our work are [23]–[34] where they focused on the optimization of content placement for cache-enabled small-cell networks. Firstly, [23] studied the optimal caching and user association strategy for a small cell network with a macro cell and multiple cache-enabled small-cell BSs (SBSs) which was similar to ours. However, multicast transmission and collaboration at the BSs were not considered with also the limits of storing entire files and homogenous file popularity.

By using file partitioning and network coding, storing subfiles in the caches instead of storing entire files has been well recognized as an effective way to improve content diversity. The optimal uncoded and coded data allocation strategies with the minimum expected costs were studied in [24], where only one single file was considered ignoring the diversity of the required file library in practice. In [25], both the analysis and optimization were extended to the multiple files scenario with two partition-based caching designs studied for a large scale successive interference cancellation (SIC)-enabled wireless network. In [26], MDS coded caching was considered with homogeneous network settings, i.e., same file sizes, cache sizes and file popularity for all the cells, which gave rise to identical content placement in all cells. Any cache miss was dealt with by separate costly unicast transmissions via the backhaul.

In addition to the studies on caching strategies using multiple unicast transmissions to serve the requests mentioned above, multicasting transmission at BSs to serve the requests for the same file simultaneously has been explored to support massive content delivery over wireless networks. In [27], joint throughput-optimal caching and scheduling algorithms were developed to maximize the service rates with both elastic and inelastic requests. For inelastic services, optimal mutlicasting scheduling was discussed while unicast communication was assumed for elastic requests. In another work [28], the authors studied uncoded multicast-aware caching in delay tolerant networks, with the assumption that consecutive requests for the same file within a multicast period can be served by a single multicast transmission. Although heterogeneous settings were assumed, no extra challenges were brought in this case since the discrete optimization problem was solved in a rather heuristic and exhaustive manner with all the possible joint user request profiles fully listed and calculated which limits its usage in large scale networks and coded caching scenarios. In our previous work [29], although coded multicast-aware caching was proposed, the research was limited to the partly heterogeneous settings of distinct cache and file sizes but homogeneous file popularity and numbers of users in all the cells. Beisdes caching design, [30], [31] offer performance analysis towards caching and multicasting for single-tier and multi-tier heterogenous networks (HetNets), respectively. Although they provide some content diversity, the assumptions made in [30] and [31] greatly limit the full usage of this diversity. For instance, the file library for the BSs in the same tier to cache from is actually the same while those for BSs in different tiers are mutually exclusive. The identical caching in the macro-tier, the random caching design with the same probability distribution in the pico-tier as well as the uncoded caching limitation of storing entire files altogether lead to this issue. Another main difference is that they focused more on multicast transmission between caches and users while we also exploit the multicast opportunities for delivering the uncached content.

While the works mentioned above are offline schemes with limited cache sizes, an online cooperative caching scheme with infinite cache capacity was presented in [32]. In this case, the energy consumption for content updating in the caches was considered which can be ignored in offline schemes in a long-term time scale. Due to the fact that the previous content placement and the current user demands were given and the caching policies for different files were mutually independent, the formulated problem was actually linear and therefore could be easily solved. Subsequently in [33], the study was extended to the joint design of caching, routing and interference management with perfect user request information.

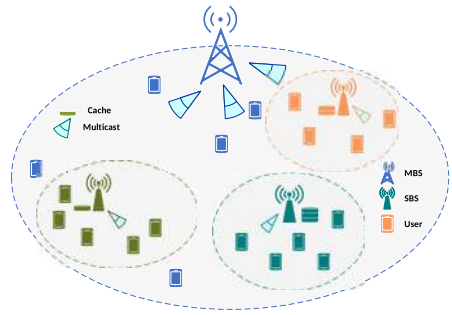
Finally in [34], an in-network cooperative caching scheme was proposed assuming that the cooperative SBSs were connected to the same service gateway to share cached content. It was assumed that the costs for fetching content from any of the cooperative SBSs were identical and so did the costs for fetching content from the content provider to the SBSs. In that effort, a cooperative caching utility maximization problem was decomposed into a number of sub-problems in different network domains and addressed by a decentralized heuristic scheme with the strong assumption of knowing the actual file demands of each user. Furthermore, the scheme is suboptimal, and the heterogeneity of the locations of the SBSs and file popularity in different cells were not well addressed.

B. Contributions

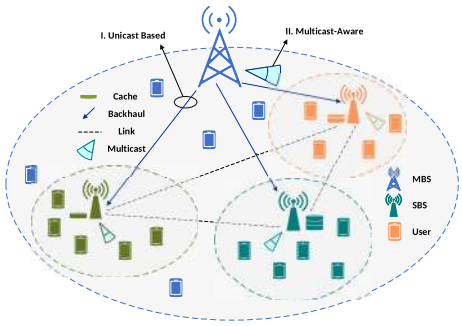
Considering the heterogeneity of cache-enabled small-cell networks, such as distinct file popularity, file sizes, cache sizes, coverages and locations of different SBSs, not only requires redesign of content placement but also cache size allocation amongst the SBSs, as mentioned in [35] and [36]. In this setup, cache size allocation and content placement in different cells will generally not be the same. Considering also the fact that file sizes may be large compared to the limited cache size in practice, files are usually split into fragments. Nevertheless, note that all of the above-mentioned works considered whole file caching except [26], [29], [33]. When the fragments are randomly selected and stored in the caches without coding, both the number of fragments in each cell and which fragments that are stored (i.e., the degree of content duplication amongst the cells), determine the backhaul load. As a result, it would be very difficult for the macro base station (MBS) to deliver the uncached content via a shared link to all the cells and unicast content delivery is therefore commonly used between the MBS and SBSs at the expense of high backhaul cost [23]–[27], [32]–[34]. On the other hand, cache content overlap among different cells would restrain cooperative caching from being effective.

In this paper, our aim is to unleash the potential of multicast-aware caching and cooperative caching by taking advantages of the inherent independence amongst the MDS coded packets for minimizing the average bakchaul rate. In summary, this paper has made the following major contributions:

- We develop offline caching schemes optimizing the long-term average performance of the cache-enabled network by estimating all possible joint user requests in different cells simultaneously without the knowledge of the actual user requests assumed in [32]–[34]. Furthermore, unlike [28], we classify the large number of possible user request profiles into several types according to their values of the associated backhual load and therefore reduce the computational complexity in terms of user request uncertainty in the analysis of multicast-aware caching. Moreover, a multicast-aware in-cluster cooperative approach is proposed suitable for large-scale networks.
- Unlike the homogenous settings considered in [26], [29], and [34], the heterogeneity of the parameters that affects the design of cache management and cooperative policy is all considered with the coordination among different SBSs and files. Also, cache size allocation is optimized subject to an overall cache capacity budget rather than uniform or an arbitrarily given heterogeneous allocation in literature.
- Furthermore, we derive the performance gains of storing coded packets over uncoded fragments in the caches and quantify the advantages of multicast-aware and cooperative caching over common caching schemes via mathematical analysis or/and simulation results. Benefited from the independence of the MDS coded packets, we combine the merits of multicast-aware caching and cooperative caching to greatly reduce the backhaul load.



(a) Multicast-aware caching



(b) Cooperative caching

Fig. 1. Cache-enabled heterogeneous small-cell networks.

II. Our Model

A. Network Model

A small cell network is considered which comprises a single MBS, and K non-overlapping small cells each consisting of a single SBS and I_k users, for the kth cell. Let $\mathcal{K} \triangleq \{1, \ldots, K\}$ denote the set of SBSs which operate in disjoint subchannels with the MBS in order to remove the impact of interference. Besides, any interference among neighboring SBSs is assumed eliminated by techniques such as enhanced inter-cell interference coordination (eICIC) or/and orthogonal multiple access [37], [38]. We assume that the MBS has access to all files in the set $\mathcal{F} \triangleq \{f_1, f_2, \ldots, f_N\}$ with respective file sizes $\mathbf{s} \triangleq [s_1, s_2, \ldots, s_N]$ while the SBSs have limited cache capacities that are subject to a network-wide total cache capacity budget M. We let M_k denote the cache capacity for SBS k, with $M_k \leq \sum_{j=1}^{N} s_j$. SBSs can push the cached packets to the users when requested while the uncached parts have to be delivered to the SBSs via backhaul from the MBS (or cooperative SBSs in the case of cooperative caching). Note that the users located outside of any small cells can only be served by the MBS and hence are ignored when considering the backhaul requirements.

1) Multicast-Aware Caching: If this approach is used, SBSs will fetch the uncached content from the MBS via backhaul using multicast, see Fig. 1a. Based on the file popularity information, we obtain the optimal content placement to minimize the average backhaul load for all possible user request profiles with the overall cache capacity budget.

2) Cooperative Caching: As shown in Fig. 1b, neighboring SBSs can be connected to each other via high-capacity links to share their cached content in different cells collaboratively.

In this scheme, the uncached content can be fetched from not only the MBS via backhaul but also the cooperative SBSs via the fronthaul links. Considering the different costs for fetching content from the MBS and the neighboring SBSs, we adopt the concept of user attrition (UA) cost introduced in [32] to evaluate the performance of the cooperative caching scheme.[1] Cache content placement and the policy for SBS cooperation are to be jointly optimized to minimize the UA cost. Unless stated otherwise, this scheme uses unicast for content delivery.

3) Multicast-Aware Cooperative Caching: In this approach, multicast-based content delivery and content sharing amongst neighboring SBSs are combined with the aid of MDS codes. In contrast to conventional cooperative caching, multicasting is applied by the MBS to deliver content to the SBSs requesting the same file simultaneously, see Case II of Fig. 1b.

B. MDS Coding

MDS codes are employed to construct pieces of a file that can be put back together to recover the file. They are particularly suitable for our settings of multicast-aware caching and cooperative caching in which the cached content in different cells needs to be coordinated. Compared to the case of storing uncoded fragments, MDS codes bring a unique benefit that the coded packets are all independent from each other so that a certain number of randomly drawn packets will be sufficient to recover the file. This allows us to use only the number of packets stored in each cell, instead of the details of the packets, to derive the backhaul load, simplifying the analysis.

We parametrize MDS codes by (l_j, n_j) such that file j is cut into n_j fragments and then coded into l_j independent packets by MDS. Any n_j packets can rebuild the entire file.

Considering that the kth SBS caches $m_{k,j}$ coded packets of file j, we let $\mathbf{m}^j \triangleq [m_{1,j}, m_{2,j}, \ldots, m_{K,j}]$ be the content placement vector for file j. For multicast-aware caching, to ensure that the uncached packets delivered from the MBS are totally different from the ones cached in local servers, file j should be coded into at least

$$l_j = \underbrace{\sum_{k=1}^{K} m_{k,j}}_{\text{unique packets cached in SBSs}} + \underbrace{n_j - \min_{k\in\{1,\ldots,K\}} m_{k,j}}_{\text{unique packets delivered via backhual}} \quad \text{packets.}$$

For unicast and multicast-aware cooperative caching scenarios, the total number of packets has to be at least

$$l_j = \underbrace{\sum_{k=1}^{K} m_{k,j}}_{\text{unique packets cached in SBSs}} + \underbrace{n_j - \min_{k\in\{1,\ldots,K\}} \sum_{t=1}^{K} x_{k,j}^{t}}_{\text{unique packets delivered via backhual}},$$

where $x_{k,j}^{t}$ denotes the number of packets delivered from SBS t to SBS k to serve the requests for file j so that there is no content overlap in both content sharing process amongst the cooperative SBSs and content delivery phase at the MBS.

[1]UA cost is the overall cost for fetching content from an external storage.

C. File Popularity Profile

Note that users in different cells may have different preferences towards the files. The most popular file in one cell may receive least attentions from another cell. It is thus better to consider local file popularity in each cell rather than the global popularity in the entire network which is often the case in the literature. Without loss of generality, here we assume that the file popularity in each cell obeys Zipf's distribution but with unique skewness parameter and popularity rank. According to the Zipf's law, the frequency for file j to be requested by each user in cell k can then be written as [39]

$$p_{k,j} = \frac{\left(1/\lambda_{k,j}^{\gamma_k}\right)}{\sum_{i=1}^{N}(1/i^{\gamma_k})}, \quad \forall k, j, \tag{1}$$

where γ_k is the skewness in cell k reflecting the concentration of the popularity distribution and $\lambda_{k,j}$ denotes the rank of the popularity of file j in cell k. For instance, $\lambda_{k,j} = 1$ means file j is the most popular file in cell k. Hence, the probability of file j not being requested by the users in cell k is

$$\alpha_{k,j} = (1 - p_{k,j})^{I_k}, \quad \forall k, j. \tag{2}$$

Thus, the probability for file j being requested by at least one of the users in cell k will be $1 - \alpha_{k,j}$.

III. Multicast-Aware Caching

The aim is to minimize the average backhaul load for all possible user request profiles, meaning that content placement should be done to satisfy different requests for all the cells simultaneously with a single multicast transmission instead of multiple unicast transmissions to each SBS separately.

A. Problem Formulation

Different from the literature where the knowledge of the actual requests from the cells was usually assumed, we analyze all possible request profiles and their probabilities using the learned file popularity. Here, the joint user request profile in all the cells is focused rather than the user request profiles in individual cells. We let Π_j denote the collection of all the possible user request profiles and $\boldsymbol{\pi}_j \in \Pi_j$ denote a particular user request profile for file j in all cells. Given any user request profile $\boldsymbol{\pi}_j$, $\mathcal{K}_{\boldsymbol{\pi}_j}$ is used to denote the set of the cells where file j is required by the served users. In case that file j is requested in all the cells except cell K, we have $\boldsymbol{\pi}_j = [1, 1, \ldots, 1, 0]_{1\times K}$ where 1 means that file j is requested by users in the considered cell while 0 states that none of the users in the cell requests the file. Therefore, it follows that $\mathcal{K}_{\boldsymbol{\pi}_j} = \{1, 2, \ldots, K-1\}$ for the mentioned $\boldsymbol{\pi}_j$. The joint user request profile for all the files simultaneously can be written as $\{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N\}$. For each file j, if there are t $(\leq K)$ cells where the served users request file j, the corresponding file request profile $\boldsymbol{\pi}_j$ and the cell set $\mathcal{K}_{\boldsymbol{\pi}_j}$ may have $\binom{K}{t}$ possible combinations. In this way, we evaluate that the total number of different $\boldsymbol{\pi}_j$ and $\mathcal{K}_{\boldsymbol{\pi}_j}$ will be as high as 2^K.

The average backhaul load is defined as the average volume of the file packets requiring to be fetched from the MBS via

backhaul with a single multicast transmission in terms of all possible user request profiles. Our objective is to minimize the average backhaul load subject to the overall cache capacity constraint. Mathematically, that is,

$$\min_{\{m_{k,j}\}} \sum_{\{\boldsymbol{\pi}_1,\ldots,\boldsymbol{\pi}_N\}} \sum_{j=1}^{N} \left(1 - \min_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} \frac{m_{k,j}}{n_j}\right) s_j P_r(\{\boldsymbol{\pi}_1,\ldots,\boldsymbol{\pi}_N\}) \tag{3a}$$

$$\text{s.t.} \sum_{k=1}^{K}\sum_{j=1}^{N} \frac{m_{k,j}}{n_j} s_j \le M, \tag{3b}$$

$$0 \le m_{k,j} \le n_j, \quad \forall k, j, \tag{3c}$$

where $P_r(\{\boldsymbol{\pi}_1,\ldots,\boldsymbol{\pi}_N\})$ denotes the joint probability that a certain user request profile for all the files, i.e., $\{\boldsymbol{\pi}_1,\ldots,\boldsymbol{\pi}_N\}$ appears. Since there are multiple cells, users and also requested files, the required analysis and calculation of the joint probabilities would be rather complex. To this end, the following lemma is used to simplify the objective function in (3a).

Lemma 1: Based on the fact that the backhaul load for a particular file j only relies on $\boldsymbol{\pi}_j$ regardless of $\{\boldsymbol{\pi}_i\}_{i\neq j}$, the average backhaul rate in (3a) can be rewritten as

$$R_{\text{multicast}}^{\text{MDS}} = \sum_{j=1}^{N} \sum_{\boldsymbol{\pi}_j\in\Pi_j} \left(1 - \min_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} \frac{m_{k,j}}{n_j}\right) s_j P_r(\boldsymbol{\pi}_j). \tag{4}$$

where $P_r(\boldsymbol{\pi}_j)$ is the probability that $\boldsymbol{\pi}_j$ appears.

Proof: See [29, Appendix A]. ∎

The following lemma exploits the relationships among the elements in $\mathbf{m}^j$ to express $R_{\text{multicast}}^{\text{MDS}}$ in closed form. Let $r_{k,j}$ be the rank of the value of $m_{k,j}$ among those of all the elements in $\mathbf{m}^j$. For instance, $r_{k,j}=1$ means $m_{k,j}$ is the smallest in $\mathbf{m}^j$ while $r_{k,j}=K$ states that $m_{k,j}$ is the largest.

Lemma 2: The backhaul load in (3a) can be rewritten as

$$R_{\text{multicast}}^{\text{MDS}} = \sum_{j=1}^{N}\sum_{k=1}^{K} \left(1 - \frac{m_{k,j}}{n_j}\right) s_j \ (1-\alpha_{k,j}) \prod_{t\in\mathcal{T}_{k,j}} \alpha_{t,j}, \tag{5}$$

in which $\mathcal{T}_{k,j}$ denotes the collection of cells storing no more packets of file j than cell k, i.e., $\mathcal{T}_{k,j} = \{t | r_{t,j} < r_{k,j}\}$.

Proof: See Appendix A. ∎

B. Comparison

As a comparison, in the typical unicast case, the backhaul rate for storing uncoded fragments directly or the MDS coded packets would have been given by

$$R_{\text{unicast}} = \sum_{j=1}^{N}\sum_{k=1}^{K} \left(1 - \frac{m_{k,j}}{n_j}\right) s_j (1-\alpha_{k,j}). \tag{6}$$

It can be observed in (5) and (6) that additional multipliers $0 < \prod_{t\in\mathcal{T}_{k,j}} \alpha_{t,j} \le 1, \forall k, \forall j$ appear after using multicast transmission at the MBS in the content delivery phase, and hence bring a global gain, i.e., $R_{\text{multicast}}^{\text{MDS}} < R_{\text{unicast}}$ [5]. On the other hand, it is worth pointing out that storing MDS coded packets has advantages over uncoded segments in the case of multicast-aware caching for minimizing the average backhaul rate. We assume that cell k stores $m_{k,j}$ different fragments randomly drawn from the n_j fragments *equiprobably*, and all fragments except the ones stored in all the cells requesting the particular file have to be sent from the MBS. Therefore,

$$R_{\text{multicast}}^{\text{uncoded}} = \sum_{j=1}^{N} \sum_{\boldsymbol{\pi}_j\in\Pi_j} \left(1-\rho_{\boldsymbol{\pi}_j}\right) s_j P_r(\boldsymbol{\pi}_j), \tag{7}$$

where $\rho_{\boldsymbol{\pi}_j}$ denotes the probability of a certain fragment of file j being stored in all the cells requesting the file given by

$$\rho_{\boldsymbol{\pi}_j} = \prod_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} \frac{\binom{n_j-1}{m_{k,j}-1}}{\binom{n_j}{m_{k,j}}} = \prod_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} \frac{m_{k,j}}{n_j}. \tag{8}$$

Since $\frac{m_{k,j}}{n_j} \le 1, \forall k$, it holds true that $\rho_{\boldsymbol{\pi}_j} \le \min_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} \frac{m_{k,j}}{n_j}$. Thus, we derive that $R_{\text{multicast}}^{\text{MDS}} \le R_{\text{multicast}}^{\text{Uncoded}}$. A rigorous proof has been provided in our previous work [39].

C. Optimization

Defining $q_{k,j} \triangleq \frac{m_{k,j}}{n_j}$ and using (5), (3) can be recast into

$$\min_{\{q_{k,j}\}} \sum_{j=1}^{N}\sum_{k=1}^{K} \left(1-q_{k,j}\right) s_j \ (1-\alpha_{k,j}) \prod_{t\in\mathcal{T}_{k,j}} \alpha_{t,j} \tag{9a}$$

$$\text{s.t.} \sum_{k=1}^{K}\sum_{j=1}^{N} q_{k,j} s_j \le M, \tag{9b}$$

$$0 \le q_{k,j} \le 1, \quad \forall k, j. \tag{9c}$$

Unfortunately, before $\{q_{k,j}\}$ are obtained, it is impossible to know the ranks $\{r_{k,j}\}$, or $\mathcal{T}_{k,j}$. To tackle this, we sort the elements of $\mathbf{q}^j, \forall j$ in an ascending order and define the sorted variables as $\mathbf{g}^j \triangleq [g_{1,j},\ldots,g_{K,j}], \forall j$. To illustrate the relationships between $\mathbf{q}^j$ and $\mathbf{g}^j$, a new matrix $\mathbf{Y} \triangleq [y_{t,j}^k]_{K\times N\times K}$ with $y_{t,j}^k \in \{0,1\}$ is defined such that

$$q_{k,j} = \sum_{t=1}^{K} g_{t,j} y_{t,j}^k. \tag{10}$$

If $q_{k,j}$ is the tth lowest in $\mathbf{q}^j$, i.e., $r_{k,j}=t$, we let $y_{t,j}^k = 1$ and $y_{\bar{t},j}^k = 0, \forall \bar{t} \neq t$. Note that the ranks are assumed to be unique integers even if there are several elements of $\mathbf{q}^j$ equal to each other. The characteristics of $\{y_{t,j}^k\}$ are concluded in the following constraints (11e)–(11g). Now, (9) becomes

$$\min_{\{g_{t,j}\},\{y_{t,j}^k\}} \sum_{j=1}^{N}\sum_{t=1}^{K} \left(1-g_{t,j}\right) s_j \varphi_{t,j} \tag{11a}$$

$$\text{s.t.} \sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{t=1}^{K} g_{t,j} y_{t,j}^k s_j \le M, \tag{11b}$$

$$g_{t,j} \le g_{t+1,j}, \quad \forall t < K, \text{ and } \forall j, \tag{11c}$$

$$0 \le g_{t,j} \le 1, \quad \forall t, j, \tag{11d}$$

$$\sum_{t=1}^{K} y_{t,j}^k = 1, \quad \forall k, j, \tag{11e}$$

$$\sum_{k=1}^{K} y_{t,j}^k = 1, \quad \forall t, j, \tag{11f}$$

$$y_{t,j}^k \in \{0,1\}, \quad \forall t, j, k, \tag{11g}$$

where $\varphi_{t,j}$ is the probability that $100g_{t,j}\%$ of file j requires delivery from the MBS via backhaul. Define a new group of variables $\{\sigma_t\}$ satisfying $q_{\sigma_t,j} = g_{t,j}$ as the indices mapping $g_{t,j}$ to $q_{\sigma_t,j}$. For instance, $\sigma_t = 1$ states that $q_{1,j}$ ranks the tth in $\mathbf{q}^j$, i.e., $q_{1,j} = g_{t,j}$. And we can then obtain the expression of $\varphi_{t,j}$ given by $\varphi_{t,j} = (1-\alpha_{\sigma_t,j})\prod_{v=1}^{t-1}\alpha_{\sigma_v,j}$ based on (9a) and the definition of $\mathbf{g}^j$. Utilizing (10), it holds true that $y_{t,j}^{\sigma_t} = 1$. Hence, $\varphi_{t,j}$ can be further rewritten as

$$\varphi_{t,j} = \left[\sum_{k=1}^{K}\left(1-\alpha_{k,j}\right)y_{t,j}^{k}\right]\prod_{v=1}^{t-1}\left[\sum_{k=1}^{K}\left(\alpha_{k,j}y_{v,j}^{k}\right)\right], \quad \forall t > 1 \tag{12}$$

with $\varphi_{1,j} = \sum_{k=1}^{K}\left(1-\alpha_{k,j}\right)y_{1,j}^{k}$.

Due to the coupling among the variables in the constraints as well as the objective function, (11) is a mixed integer nonlinear program (MINLP) and is difficult to deal with. The expression of $\varphi_{t,j}$ also makes it too complex to be linearized. As such, reformulation is done here to simplify the constraints.

Lemma 3: Based on the characteristics of $\{y_{t,j}^k\}$, the overall cache capacity constraint in (11b) can be re-expressed as $\sum_{t=1}^{K}\sum_{j=1}^{N} g_{t,j}s_j \le M$. Hence, (11) can be rewritten as

$$\min_{\{g_{k,j}\},\{y_{t,j}^k\}} \quad \sum_{j=1}^{N}\sum_{t=1}^{K}\left(1-g_{t,j}\right)s_j\varphi_{t,j} \tag{13a}$$

$$\text{s.t.} \quad \sum_{t=1}^{K}\sum_{j=1}^{N} g_{t,j}s_j \le M, \tag{13b}$$

$$(11c)\text{–}(11g), \tag{13c}$$

with the optimal allocated cache sizes given by

$$M_k = \sum_{j=1}^{N}\sum_{t=1}^{K} g_{t,j}y_{t,j}^{k}s_j, \quad \forall k. \tag{14}$$

Proof: According to (10), it can be easily proved that (14) holds. Then utilizing the constraint (11f), we obtain

$$\begin{aligned}\sum_{k=1}^{K} M_k &= \sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{t=1}^{K} g_{t,j}y_{t,j}^{k}s_j, \\ &= \sum_{j=1}^{N}\sum_{t=1}^{K} g_{t,j}\left(\sum_{k=1}^{K} y_{t,j}^{k}\right)s_j = \sum_{j=1}^{N}\sum_{t=1}^{K} g_{t,j}s_j. \end{aligned} \tag{15}$$

Hence, we get (13b), which completes the proof. ∎

After utilizing *Lemma 3*, $\{g_{t,j}\}$ and $\mathbf{Y}$ are now decoupled in the constraints of (13). To proceed, we firstly fix $\{g_{t,j}\}$ and optimize $\mathbf{Y}$. The problem of interest is given by

$$\mathcal{P}(\{g_{t,j}\}): \quad \min_{\{y_{t,j}^k\}} \quad \sum_{j=1}^{N}\sum_{t=1}^{K}\left(1-g_{t,j}\right)s_j\varphi_{t,j} \tag{16a}$$

$$\text{s.t. } (11e)\text{–}(11g), \tag{16b}$$

with $\{g_{t,j}\}$ satisfying (11c)–(11d) and (13b). Obviously, $\{y_{t,j}^k\}$ are independent with each other in different files in problem (16). As a result, we can separate the problem into a number of sub-problems with regard to different file j, e.g.,

$$\mathcal{P}_j(\{g_{t,j}\}): \quad \min_{\{y_{t,j}^k\}} \quad \sum_{t=1}^{K}\left(1-g_{t,j}\right)\varphi_{t,j} \tag{17a}$$

$$\text{s.t.} \quad \sum_{t=1}^{K} y_{t,j}^{k} = 1, \quad \forall k, \tag{17b}$$

$$\sum_{k=1}^{K} y_{t,j}^{k} = 1, \quad \forall t, \tag{17c}$$

$$y_{t,j}^{k} \in \{0,1\}, \quad \forall t,k. \tag{17d}$$

The coupling and complexity of $\varphi_{t,j}$ makes it intractable to find the optimal $\{y_{t,j}^k\}$ even when $\{g_{t,j}\}$ are given. To tackle this problem, we analyze the impact of $\{y_{t,j}^k\}$ on the objective function based on the characteristics of $\{g_{t,j}\}$ and $\{y_{t,j}^k\}$, and infer the relations among $\{y_{t,j}^k\}$ and the probabilities $\{\boldsymbol{\alpha}^j\}$. For illustrative purposes, we let $\boldsymbol{\alpha}^j \triangleq [\alpha_{1,j},\alpha_{2,j},\ldots,\alpha_{K,j}]$, rearrange the elements in $\boldsymbol{\alpha}^j$ in a descending order and define the new vector as $\boldsymbol{\beta}^j \triangleq [\beta_{1,j},\beta_{2,j},\ldots,\beta_{K,j}]$. Let $\{\theta_k\}$ reflect the one-to-one correspondence between the elements of $\boldsymbol{\beta}^j$ and $\boldsymbol{\alpha}^j$ satisfying $\beta_{k,j} = \alpha_{\theta_k,j}, \forall k$. Meanwhile, $\boldsymbol{\alpha}^j$, $\boldsymbol{\beta}^j$, and $\{\theta_k\}$ are all known. The result is given in the following lemma.

Lemma 4: The optimal probability $\varphi_{t,j}^*$ would be $\varphi_{t,j}^* = \left(1-\beta_{t,j}\right)\prod_{v=1}^{t-1}\beta_{v,j}$. Accordingly, the optimal $\{y_{t,j}^k\}$ to problem (17) are given by

$$y_{t,j}^{k} = \begin{cases} 1, & \text{if } k = \theta_t, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Proof: See Appendix B. ∎

Since *Lemma 4* holds true for all the files, (13) becomes

$$\min_{\{g_{t,j}\}} \quad \sum_{j=1}^{N}\sum_{t=1}^{K}\left(1-g_{t,j}\right)s_j\left(1-\beta_{t,j}\right)\prod_{v=1}^{t-1}\beta_{v,j} \tag{19a}$$

$$\text{s.t. } (13b)\text{–}(13c), \tag{19b}$$

which is convex and hence can be easily solved by well known solvers, e.g., CVX [40]. Then substituting (18) into (14), the optimal cache capacities in each cell can be rewritten as

$$M_k = \sum_{j=1}^{N} g_{t,j}s_j|_{\theta(t)=k}, \quad \forall k, \tag{20}$$

with the optimal content placement given by

$$q_{k,j} = g_{t,j}|_{\theta(t)=k}, \quad \forall k,j. \tag{21}$$

In the proposed multicast-aware caching scheme, we classify the large number of possible user request profiles into several types according to the values of the associated backhual load. By doing so, we reduce the computational complexity in terms of user request uncertainty massively from $O(N^K)$ to $O(KN)$ to obtain the optimal solution.

IV. Cooperative Caching

In this section, we consider that the SBSs can fetch content from the neighboring SBSs via some high capacity links and study the optimal cooperative caching policy among the SBSs.

Note that the independence amongst the MDS coded packets cached in all the cells almost surely guarantees that the shared contents are always non-overlapping.

A. Problem Formulation

Cooperative caching consists of three phases:
(i) the content placement phase,
(ii) the content sharing phase among the SBSs, and
(iii) the content delivery phase from the MBS via backhaul.
Note that in the content delivery phase, we assume that unicast is used by the MBS to sent uncached content to the SBSs.

Since backhual load is unable to provide sufficient insight about the impact of cooperative content sharing on reducing the backhual requirements, here we utilize user attrition (UA) cost, i.e., the overall cost for fetching content from an external storage, to evaluate the performance of the cooperative caching schemes. To further eliminate the redundancy, we assume that the SBSs can selectively deliver part of the packets from their own caches to the requested SBS rather than the whole of the cached packets. The amounts of shared content among the cooperative SBSs are defined as $\mathbf{X} = \{x_{k,j}^t\}_{K\times N\times K}$ where $x_{k,j}^t$ denotes the number of packets delivered from SBS t to SBS k for file j. Thus, we let f_k^t be the associated unit cost when SBS k fetches unit data (e.g., per MB) from SBS t and f_k^M be the cost for delivering unit data to SBS k from MBS.

The UA costs are modeled as the products of the data loads of the BSs and the associated unit costs [32]. Furthermore, it is assumed that the unit costs are proportional to the square of the minimum distances between the associated BSs with the unit cost coefficients defined as f_0 and f_0^M, respectively, according to [24], [28], and [32]. Note that $\{f_k^t\}$ must satisfy the triangle inequality, i.e., $f_k^t \le f_l^t + f_k^l$, and the cost for fetching content from local storage can be ignored, i.e., $f_k^k = 0, \forall k$. Moreover, the UA costs for fetching content from the MBS via backhaul are usually higher than those caused by the cooperation between the SBSs due to proximity.

Instead of focusing on the backhaul load, our objective here is to minimize the average UA cost, i.e., the cost of fetching content from external storage, subject to a given overall cache capacity constraint by optimizing the cache content placement and cooperation policy jointly. In this case, the expected UA cost defined as $C_{\text{coop}}^{\text{MDS}}$ can be written as

$$C_{\text{coop}}^{\text{MDS}} = \sum_{j=1}^{N}\sum_{k=1}^{K}\left[\left(1-\min\left(1,\sum_{t=1}^{K}\frac{x_{k,j}^t}{n_j}\right)\right)f_k^M + \sum_{t=1}^{K}\frac{x_{k,j}^t}{n_j}f_k^t\right]s_j(1-\alpha_{k,j}). \quad (22)$$

Hence, the problem of interest is given by

$$\min_{\{m_{k,j}\},\{x_{k,j}^t\}} \quad C_{\text{coop}}^{\text{MDS}} \quad (23a)$$

$$\text{s.t.} \sum_{k=1}^{K}\sum_{j=1}^{N}\frac{m_{k,j}}{n_j}s_j \le M, \quad (23b)$$

$$0 \le m_{k,j} \le n_j, \quad \forall k, j, \quad (23c)$$

$$0 \le x_{k,j}^t \le m_{t,j}, \quad \forall k, j, t, \quad (23d)$$

where the cache size allocation problem is merged into the optimization of the content placement as mentioned in *Lemma 3*. Apparently, $x_{k,j}^k = m_{k,j}, \forall k, j$ holds true in (23).

B. Comparison

The significance of adopting MDS codes is to avoid content overlap among the fragments stored in different caches, hence reducing the average UA cost. Suppose that SBS k stores $m_{k,j}$ different fragments randomly drawn among the n_j fragments and $x_{t,j}^k$ of the $m_{k,j}$ fragments are randomly selected to be sent to SBS t. It is difficult to ensure that the fragments from the neighboring cells are always mutually exclusive. Thus, both the number of fragments stored in local cache and sent to other cells and which fragments being cached and shared contribute in deciding the backhaul rate and the average UA cost.

Lemma 5: Given any cooperative caching policy satisfying constraints (23b)–(23d), the UA cost in the coded scenario is always lower than the associated cost in the uncoded scenario defined as $C_{\text{coop}}^{\text{uncoded}}$, i.e., $C_{\text{coop}}^{\text{MDS}} \le C_{\text{coop}}^{\text{uncoded}}$.

Proof: See Appendix C. ∎

C. Optimization

We can tackle (23) by proving that the optimal cooperative caching policy always satisfies $\sum_{t=1}^{K}\frac{x_{k,j}^t}{n_j} \le 1, \forall k, j$. Letting $(\{\tilde{x}_{k,j}^t\}, \{\tilde{m}_{k,j}\})$ be the optimal solution to (23) with at least a group of (k^*, j^*) satisfying $\sum_{t=1}^{K}\frac{\tilde{x}_{k^*,j^*}^t}{n_j} > 1$, we can always find some $(\{x_{k,j}^t\}, \{\tilde{m}_{k,j}\})$ with $x_{k,j}^t = \tilde{x}_{k,j}^t, \forall\ (k, j, t) \ne (k^*, j^*, t)$ and $\sum_{t=1}^{K}\frac{\tilde{x}_{k^*,j^*}^t}{n_j} = 1$ which satisfy all the constraints in (23) while demanding the same cost from backhaul but a lower cost from content sharing among the cooperative SBSs. Consequently, the average UA cost is given by

$$C_{\text{coop}}^{\text{MDS}} = \sum_{j=1}^{N}\sum_{k=1}^{K}\left[\left(1-\sum_{t=1}^{K}z_{k,j}^t\right)f_k^M + \sum_{t=1}^{K}z_{k,j}^t f_k^t\right] \times s_j(1-\alpha_{k,j}), \quad (24)$$

where we let $q_{k,j} = \frac{m_{k,j}}{n_j}$ and $z_{k,j}^t = \frac{x_{k,j}^t}{n_j}$. Problem (23) can then be rewritten as

$$\min_{\{q_{k,j}\},\{z_{k,j}^t\}} \quad (24) \quad (25a)$$

$$\text{s.t.} \sum_{k=1}^{K}\sum_{j=1}^{N} q_{k,j}s_j \le M, \quad (25b)$$

$$0 \le q_{k,j} \le 1, \quad \forall k, j, \quad (25c)$$

$$\sum_{t=1}^{K} z_{k,j}^t \le 1, \quad \forall k, j, \quad (25d)$$

$$0 \le z_{k,j}^t \le q_{t,j}, \quad \forall k, j, t, \quad (25e)$$

which is linear and can easily be solved using, e.g., CVX.

For comparison, the average UA cost in the unicast based non-cooperative caching scenario is given by

$$C_{\text{noncoop}}^{\text{unicast}} = \sum_{j=1}^{N}\sum_{k=1}^{K}\left(1-q_{k,j}\right) f_k^M s_j(1-\alpha_{k,j}). \quad (26)$$

As $f_k^t \le f_k^M$ and $z_{k,j}^k = q_{k,j}, \forall k,t,j$, we have

$$C_{\text{coop}}^{\text{MDS}} \le \sum_{j=1}^{N}\sum_{k=1}^{K}\left(1-\sum_{t=1}^{K} z_{k,j}^t + \sum_{t\neq k} z_{k,j}^t\right) \times f_k^M s_j(1-\alpha_{k,j}) \le C_{\text{noncoop}}^{\text{unicast}}. \quad (27)$$

V. Multicast-Aware Cooperative Caching

In this section, a compound caching policy named multicast-aware cooperative caching is proposed to take the advantages of both multicasting at the MBS and collaboration among the SBSs. Global optimal caching scheme is proposed for small scale networks followed by the multicast-aware in-cluster cooperative caching scheme developed particularly for the large scale networks.

A. Small Scale Networks

Lemma 6: In case of multicast-aware cooperative caching, the UA cost can be written as

$$C_{\text{mult,coop}}^{\text{MDS}} = \sum_{j=1}^{N}\left[\sum_{\boldsymbol{\pi}_j\in\Pi_j}\left(1-\min_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}}\sum_{t=1}^{K} z_{k,j}^t\right)\max_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} f_k^M \times P_r(\boldsymbol{\pi}_j) + \sum_{k=1}^{K}\sum_{t=1}^{K} z_{k,j}^t f_k^t(1-\alpha_{k,j})\right] s_j. \quad (28)$$

Proof: See Appendix D. ∎

The average UA cost minimization problem is

$$\min_{\{q_{k,j}\},\{z_{k,j}^t\}} C_{\text{mult,coop}}^{\text{MDS}} \quad \text{s.t. (25b)–(25e).} \quad (29)$$

We recognize that similar content in different cells is preferred for multicast-aware caching while for cooperative caching the cached content in different cells should be mutually exclusive. The use of MDS codes strikes a balance in the combination. It is worth pointing out that multicast-aware cooperative caching brings additional multicast gain in most cases in terms of minimizing the long term average UA cost considering the large numbers of BSs, files, and user request profiles while unicast content delivery might only be preferred in rare extreme cases, e.g., when only a few cells with steeply graded unit costs require the same file. To eliminate the impact of these special cases, a new group of binary variable can be introduced to identify which content delivery strategy is preferred for each user request profile in the case of small scale networks.

Lemma 7: Given any mutlicast-aware cooperative caching policy $(\{q_{k,j}\},\{z_{k,j}^t\})$ satisfying the constraints in (29), the UA cost in the coded scenario is always much lower than that in the uncoded case, i.e., $C_{\text{mult,coop}}^{\text{MDS}} \le C_{\text{mult,coop}}^{\text{uncoded}}$.

Proof: See Appendix E. ∎

To solve (29), we resort to a greedy algorithm by listing all possible user request profiles for each file. Furthermore, a number of new variables and constraints need to be added to linearize the function $\min(\cdot)$. That is, for any user request profile $\boldsymbol{\pi}_j$, we introduce a new variable $\xi_{\boldsymbol{\pi}_j}$ subject to the constraints, i.e., $(0 \le \xi_{\boldsymbol{\pi}_j} \le \sum_{t=1}^{K} z_{k,j}^t, \forall k \in \mathcal{K}_{\boldsymbol{\pi}_j})$, to replace $\min_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}}\sum_{t=1}^{K} z_{k,j}^t$ in (28). Since (29) can be linearized, general solvers can be employed to solve it for small-scale networks. However, in practical scenarios with dozens of BSs and thousands of files, the greedy approach is not viable.

B. Large Scale Networks

In order to reduce the complexity in large scale networks, we propose a multicast-aware in-cluster cooperative caching scheme by decomposing a macro cell into a series of annular regions $\{\mathcal{C}^u, \forall u \in [1,U]\}$ with their radii between $R_u \pm \Delta R_u$ $(\Delta R_u \ll R_u)$. In each annulus, the neighboring SBSs form a number of disjoint clusters defined as $\{\mathcal{S}_1^u, \mathcal{S}_2^u, \ldots, \mathcal{S}_{L_u}^u\}$ where L_u is the number of clusters in the uth annulus. Let $|\mathcal{S}_l^u|$ denote the number of SBSs in cluster $\mathcal{S}_l^u$. It is assumed that the SBSs in the same cluster $\mathcal{S}_l^u$ can share content over high capacity links with a cost $f_l^u = f_0\overline{d}_l^u$ where $\overline{d}_l^u$ is the average of the squares of the distances among the cooperative SBSs. The cost for retrieving content from the MBS is $f_u^M = f_0^M R_u^2$ where R_u is the radius for the uth annulus. The UA cost in cluster $\mathcal{S}_l^u$ is given by

$$C_l^u = \sum_{j=1}^{N}\left[\sum_{\boldsymbol{\pi}_{l,j}^u\in\Pi_{l,j}^u}\left(1-\min_{k\in\mathcal{K}_{\boldsymbol{\pi}_{l,j}^u}}\sum_{t\in\mathcal{S}_l^u} z_{k,j}^t\right) f_u^M P_r(\boldsymbol{\pi}_{l,j}^u) + \sum_{k\in\mathcal{S}_l^u}\sum_{t\in\mathcal{S}_l^u\setminus k} z_{k,j}^t f_l^u(1-\alpha_{k,j})\right] s_j. \quad (30)$$

Therefore, this scheme solves

$$\min_{\{q_{k,j}\},\{z_{t,j}^k\}} \sum_u\sum_l C_l^u \quad (31a)$$

$$\text{s.t.} \sum_{t\in\mathcal{S}_l^u} z_{k,j}^t \le 1, \quad \forall k\in\mathcal{S}_l^u, \forall j,\ \forall l,\ \forall u, \quad (31b)$$

$$0 \le z_{k,j}^t \le q_{t,j}, \quad \forall t,k\in\mathcal{S}_l^u,\ \forall j,\ \forall l,\ \forall u, \quad (31c)$$

$$0 \le q_{k,j} \le 1, \quad \forall k\in\mathcal{S}_l^u,\ \forall j,\ \forall l,\ \forall u, \quad (31d)$$

$$\sum_u\sum_l\sum_j\sum_{k\in\mathcal{S}_l^u} q_{k,j}s_j \le M. \quad (31e)$$

For the sake of mathematical tractability, we decompose the problem into a number of sub-problems each minimizing the UA cost for a cluster. In this case, we let $q_{k,j} = q_{l,j}^u, \forall k \in \mathcal{S}_l^u, \forall j,l,u$ and the sub-problem for cluster $\mathcal{S}_l^u$ is given by

$$\mathcal{P}(\{q_{l,j}^u\}): \min_{\{z_{t,j}^k\}} C_l^u \quad (32a)$$

$$\text{s.t.} \sum_{t\in\mathcal{S}_l^u} z_{k,j}^t \le 1, \quad \forall k\in\mathcal{S}_l^u,\ \forall j, \quad (32b)$$

$$0 \le z_{k,j}^t \le q_{l,j}^u, \quad \forall t,k\in\mathcal{S}_l^u,\ \forall j. \quad (32c)$$

Because the cost for fetching content from local cache can be ignored, it holds true that $z_{k,j}^k = q_{l,j}^u, \forall k \in \mathcal{S}_l^u$. For any given

cache composition satisfying the constraints (31b)–(31e), we find it important to understand the volume of content that is needed to be fetched from the MBS via backhaul. Let $D_l^u = \sum_{j=1}^{N} \sum_{k \in \mathcal{S}_l^u} q_{l,j}^u s_j \ (1 - \alpha_{k,j})$. Given cache composition, D_l^u is always constant and hence can be ignored. The objective function can then be further reformulated into

$$\widetilde{C}_l^u = C_l^u + D_l^u = \sum_{j=1}^{N} \left[\sum_{\boldsymbol{\pi}_{l,j}^u \in \Pi_{l,j}^u} \left(1 - \min_{k \in \mathcal{K}_{\boldsymbol{\pi}_{l,j}^u}} \lambda_{k,j} \right) \times f_u^M P_r(\boldsymbol{\pi}_{l,j}^u) + \sum_{k \in \mathcal{S}_l^u} \lambda_{k,j} f_l^u (1 - \alpha_{k,j}) \right] s_j. \quad (33)$$

where $\lambda_{k,j} = \sum_{t \in \mathcal{S}_l^u} z_{k,j}^t$ denotes the percentage of file j accessible to SBS k within the cluster and is subject to

$$0 \le \lambda_{k,j} \le 1, \quad \forall k \in \mathcal{S}_l^u, \ \forall j, \quad (34)$$

$$q_{l,j}^u \le \lambda_{k,j} \le |\mathcal{S}_l^u| q_{l,j}^u, \quad \forall t, k \in \mathcal{S}_l^u, \ \forall j. \quad (35)$$

Note that with the assumption of homogeneous content placement in the SBSs in the same cluster, this gives the overall percentage of a certain file j SBS k gets access to, i.e., $\lambda_{k,j}$. In the following, we focus on obtaining the optimal values of $\{\lambda_{k,j}\}$. Similar to the multicast-aware caching scenario, the objective function can be rewritten as

$$\widetilde{C}_l^u = \sum_{j=1}^{N} \sum_{k \in \mathcal{S}_l^u} \left[\left(1 - \lambda_{k,j}\right) f_u^M \ (1 - \alpha_{k,j}) \prod_{t \in \mathcal{T}_{k,j}} \alpha_{t,j} + \lambda_{k,j} f_l^u \ (1 - \alpha_{k,j}) \right] s_j, \quad (36)$$

where $\mathcal{T}_{k,j}$ is the set of cells satisfying $\mathcal{T}_{k,j} = \{t | r_{t,j} < r_{k,j}\}$ as in *Lemma 5*. In this case, we manage to obtain the actual relation amongst $\lambda_{k,j}, \forall k \in \mathcal{S}_l^u$ in the following lemma.

Lemma 8: Given any homogeneous cache decomposition in cluster $\mathcal{S}_l^u$, it holds true that the optimal percentages for file j accessible to the SBSs within the cluster either at local cache or from the cooperative SBSs are always the same regardless of the distinct probabilities for file j being requested by users in different cells, i.e., $\lambda_{k,j} = \lambda_{t,j}, \forall k, t \in \mathcal{S}_l^u$.

Proof: See Appendix F. ∎

According to *Lemma 8*, we let $\lambda_{k,j} = \lambda_{l,j}^u, \forall k \in \mathcal{S}_l^u$. The associated UA cost in (30) can be rewritten as

$$C_l^u = \sum_j \left(1 - \lambda_{l,j}^u\right) f_u^M \omega_{l,j}^u s_j + \sum_{j=1} \sum_{k \in \mathcal{S}_l^u} \left(\lambda_{k,j} - q_{l,j}^u\right) \times f_l^u (1 - \alpha_{k,j}) s_j, \quad (37)$$

where $\omega_{l,j}^u$ is the probability for file j being requested by any of the users served by the SBSs in the cluster $\mathcal{S}_l^u$ given by

$$\omega_{l,j}^u = 1 - \prod_{k \in \mathcal{S}_l^u} \alpha_{k,j}, \quad \forall j, l, u. \quad (38)$$

Therefore, (31) can then be recast into

$$\min_{\{q_{l,j}^u\}, \{\lambda_{l,j}^u\}} \ \sum_u \sum_l C_l^u \quad (39a)$$

$$\text{s.t. } 0 \le \lambda_{l,j}^u \le 1, \quad \forall j, \ \forall l, \ \forall u, \quad (39b)$$

$$q_{l,j}^u \le \lambda_{l,j}^u \le |\mathcal{S}_l^u| q_{l,j}^u, \quad \forall j, \ \forall l, \ \forall u. \quad (39c)$$

$$0 \le q_{l,j}^u \le 1, \quad \forall j, \ \forall l, \ \forall u, \quad (39d)$$

$$\sum_u \sum_l \sum_j q_{l,j}^u s_j \le M. \quad (39e)$$

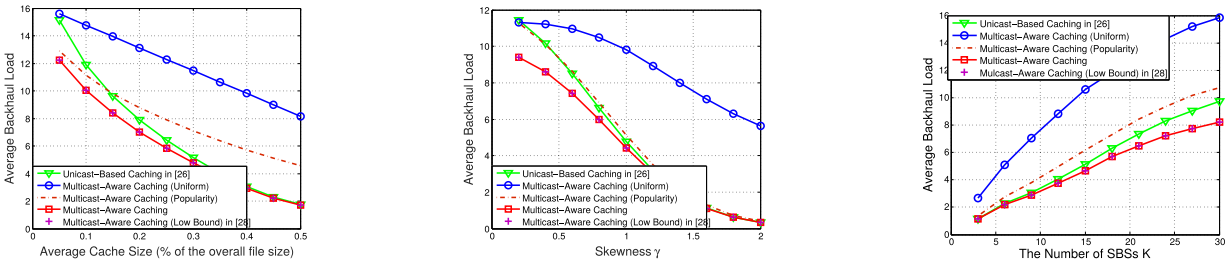
The problem is now linear with smaller sets of variables and constraints and can be solved by well-known solvers.

VI. Simulation Results

Here, we evaluate the performances of the proposed coded caching schemes in terms of the average backhaul load as well as the UA cost via computer simulations. A typical small cell network with $K = 10$ cells and $N = 100$ files is considered for the evaluation of multicast-aware caching scheme and the overall cooperative caching schemes while a large scale network with $K = 28$, $N = 1000$ is considered for in-cluster cooperative caching schemes. The MBS is located at the center of the macro cell with radius $R = 400$km while the SBSs are randomly deployed uniformly within the cell without coverage overlapping. To show clearly the capabilities for the SBSs to accommodate the files, the overall cache capacity budget is presented as the average cache size for each SBS scaled by the overall file size given by $\rho = M/K/\sum_j s_j$. Unless otherwise specified, we set $\rho = 0.25$ for multicast-aware caching and in-cluster caching schemes while $\rho = 0.05$ is assumed for overall cooperative caching schemes to ensure the participation of backhual in content delivery. The file sizes are randomly chosen uniformly within [0, 500]MB. The skewness parameters $\{\gamma_k\}$ are selected randomly within [0, 2] while the popularity ranks of the files in each cell are generated randomly. Also, the number of users in each cell is set to be ranged within [0, 10], respectively. For cooperative caching, the neighboring SBSs are linked when the distances between them are less than a given threshold. Here, we consider that two SBSs can share content in their caches when the cost for retrieving content from the other SBS is lower than that of fetching content from the MBS. The unit cost coefficients for the two routes for fetching content from external storage are set as $f_0^M = 2$ and $f_0 = 1$.

Below describes all the considered schemes.

- **Unicast-Based Caching (Non-Cooperative Caching)**: This is the unicast-based non-cooperative caching scheme with optimal cache management in [26].
- **Multicast-Aware Caching (Uniform)**: This scheme performs multicast-aware caching with uniform cache size allocation and content placement.
- **Multicast-Aware Caching (Popularity)**: This is same as above except with popularity based content placement.
- **Multicast-Aware Caching**: This refers to our *proposed* multicast-aware caching scheme with optimal cache content placement.
- **Multicast-Aware Caching (Low Bound)**: This refers to the method with optimal cache size allocation and content



(a) Impact of overall cache size M. (b) Impact of skewness γ. (c) Impact of number of SBSs K.

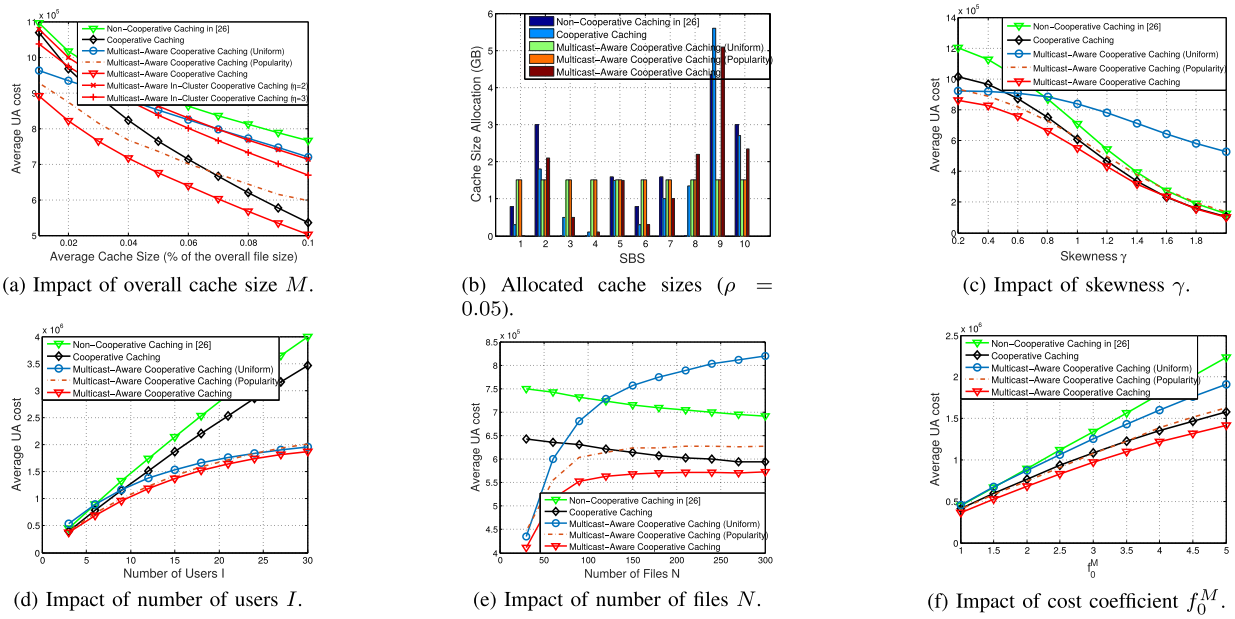
Fig. 2. The average backhaul rate of the proposed multicast-aware caching scheme versus the unicast based caching scheme and the multicast-aware caching schemes.

placement of the linear relaxed multicast aware uncoded caching problem in [28]. Notice that this is practically impossible and only serves as a lower bound.

- **Cooperative Caching**: This corresponds to our *proposed* unicast-based cooperative caching scheme with optimal cache management and cooperation policy.
- **Multicast-Aware Cooperative Caching (Uniform)**: This is the multicast-aware cooperative caching scheme that uses uniform cache size allocation and content placement.
- **Multicast-Aware Cooperative Caching (Popularity)**: Same as above except with popularity content placement.
- **Multicast-Aware Cooperative Caching**: This refers to our *proposed* multicast-aware cooperative caching with optimal cache management and cooperation policy.
- **In-Cluster Cooperative Caching**: This scheme is similar to cooperative caching except that cooperation is enabled among the SBSs in the same clusters.
- **Multicast-Aware In-Cluster Cooperative Caching**: This scheme is similar to multicast-aware cooperative caching except that multicasting and cooperation are enabled among the SBSs in the same clusters.

1) Multicast-Aware Caching: Results in Fig. 2 are provided for the proposed multicast-aware caching scheme, with different content placements, and compared with the uniform based caching scheme. Moreover, the impacts of different parameters and file profile are investigated. As can be seen in Fig. 2a, the increase of overall cache size budget leads to a decrease in backhaul rates in all the cases. Also, the proposed multicast-aware caching scheme with optimal content placement, which reaches the low bound of the multicast aware uncoded caching scheme in [28] using linear relaxation and optimal cache management at much lower commuting complexity, shows apparent advantages over the unicast based scheme as expected while the multicast-aware caching schemes with uniform and popularity based content placement show worse performances due to the naive cache management, confirming the significance of multicast transmission in content delivery as well as the centralized cache management in heterogeneous small cell networks. Similar results can be observed in Fig. 2b against the skewness parameter of the Zipf's distribution, with $\gamma = \gamma_k, \forall k$ and distinct popularity ranks for the files in different cells. The impact of the number of BSs on the backhaul rate is shown in Fig. 2c where the gain improves in denser networks. Again, the multicast-aware scheme outperforms other caching schemes.

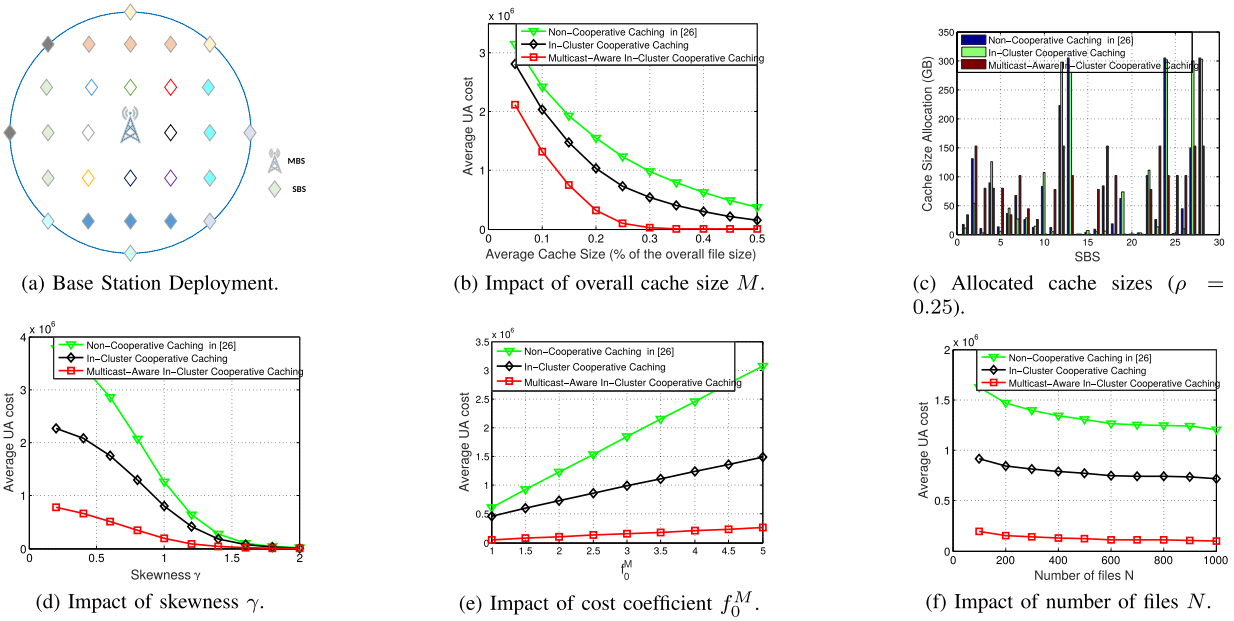
2) Cooperative Caching (Unicast and Multicast): Results in Fig. 3 compare the performance of the proposed cooperative caching schemes with that of the non-cooperative scheme in terms of the average UA cost. As can be observed, the proposed multicast-aware cooperative caching scheme shows the best performances followed by the unicast based cooperative caching scheme while the non-cooperative caching scheme yields the worst performance in all the cases. In addition, the multicast-aware cooperative caching schemes using common content placement demand higher UA costs compared with the proposed optimal multicast-aware caching scheme as expected. As we see in Fig. 3a, the UA costs decrease with the overall cache size in all cases. Apparently, the utility of cooperation in caching and multicast-aware caching reduce the average UA cost in the network dramatically. For comparison, we also present the results of multicast-aware in-cluster cooperative caching scheme with the maximum cluster size, i.e., the maximum number of SBSs in the clusters defined as η, equal to 2 and 3, respectively. Though the in-cluster caching scheme causes certain performance loss compared with the overall cooperative caching schemes, it largely reduces the computational complexity which makes it suitable for large-scale networks where overall cooperative caching schemes are unviable. Moreover, we can see in the figure that the performance gap can be narrowed by increasing the maximum cluster size η. Fig. 3b presents the cache size allocation among the SBSs using different caching schemes when $\rho = 0.05$. Results show that the optimal cache sizes for different cells are always heterogeneous as opposed to the assumption of uniform cache size allocation in many caching networks. Similar conclusions on the impacts of the skewness and the number of users to the non-cooperative case mentioned above can be drawn from Figs. 3c and 3d. Next, Figs. 3e and 3f investigate the impacts of the number of files and the cost coefficient f_0^M. As we can see in Fig. 3e, the UA cost reduction of the proposed multicast-aware cooperative caching scheme decreases with the number of files when $\rho = 0.05$ and $s_j = 250\text{MB}, \forall j$ to unicast based cooperative scheme. Finally, the impact of the ratio between the unit cost coefficients is studied in Fig. 3f where $f_0 = 1$ but f_0^M varies. Apparently, the UA cost of the non-cooperative caching scheme is proportional to f_0^M while the cooperative schemes have much better



(a) Impact of overall cache size M. (b) Allocated cache sizes ($\rho = 0.05$). (c) Impact of skewness γ.

(d) Impact of number of users I. (e) Impact of number of files N. (f) Impact of cost coefficient f_0^M.

Fig. 3. The average UA cost of the proposed cooperative caching schemes versus the non-cooperative scheme.



(a) Base Station Deployment. (b) Impact of overall cache size M. (c) Allocated cache sizes ($\rho = 0.25$).

(d) Impact of skewness γ. (e) Impact of cost coefficient f_0^M. (f) Impact of number of files N.

Fig. 4. The average UA cost of the proposed multicast-aware in-cluster cooperative caching scheme versus in-cluster cooperative caching scheme and non-cooperative caching scheme.

tolerance towards the increase of f_0^M for fetching content via backhaul.

3) Multicast-Aware and in-Cluster Cooperative Caching: Now, a large-scale small cell network with $K = 28$ cells and $N = 1000$ files is considered where the greedy algorithm for multicast-aware cooperative caching scenario is no longer efficient due to high computational complexity and hence in-cluster cooperative caching schemes are considered. Here we assume typical grid deployment of the SBSs as depicted in Fig. 4a. The MBS is located at the center of the macro cell with radius $R = 400$km and the distance between any two of the neighboring SBSs is fixed at $d = R/3$. The SBSs are divided into 4 annuli based on the distances and then the neighboring SBSs in each annulus are allocated into a number of disjoint clusters where the SBSs in the same color form a cluster. Unless stated otherwise, same parameters as before are used.

Results for the multicast-aware in-cluster caching scheme are provided in Fig. 4. We see that the multicast-aware in-cluster cooperative caching scheme achieves the best UA cost performance followed by the in-cluster cooperative caching scheme while the non-cooperative caching scheme gives the highest UA cost. Compared with that in small scale networks, the UA cost reduction becomes more obvious. The reason may

be that the network topologies are different and denser which gives rise to larger number of clusters and the average cluster size than those in the previous scenarios.

VII. Conclusions

In this paper, we considered the design of content caching and sharing for cache-enabled heterogeneous small cell networks using MDS codes under heterogeneous file and network settings. We first presented two coded caching schemes, dubbed as the multicast-aware caching and the cooperative caching schemes, for minimizing the long-term average backhaul load or the UA cost subject to the overall cache capacity constraint. In both cases, we have obtained the optimal content placement by reformulating the original problems into convex ones. A compound caching scheme, referred to as multicast-aware cooperative caching, was then proposed exploiting the independence of MDS coded packets to further reduce the backhaul requirements. In this case, a greedy algorithm can be used for small scale networks while for large scale networks a multicast-aware in-cluster cooperative caching algorithm was developed. The advantages of storing coded packets over the uncoded fragments in all the scenarios as well as the benefits of utilizing multicast-aware caching and/or cooperative caching over common caching schemes have been analyzed.

Appendix A

Firstly, we divide the possible user request profiles for each file, e.g., $\boldsymbol{\pi}_j$ into $K+1$ types defined as $\{\boldsymbol{\pi}_j^0, \boldsymbol{\pi}_j^1, \boldsymbol{\pi}_j^2, \ldots, \boldsymbol{\pi}_j^K\}$ according to the different values of the associated backhaul load (in percentage) for file j, i.e., $\{0, 1-\frac{m_{1,j}}{n_j}, 1-\frac{m_{2,j}}{n_j}, \ldots, 1-\frac{m_{K,j}}{n_j}\}$, respectively. Note that $\boldsymbol{\pi}_j^0$ states that file j is not requested by users in any of the cells, and hence backhaul is no longer needed in this case. If cell k stores the least number of packets of file j among all the cells requesting file j, i.e., $\min_{t\in\mathcal{K}_{\boldsymbol{\pi}_j}} \frac{m_{t,j}}{n_j} = \frac{m_{k,j}}{n_j}$, then the associated user request profile $\boldsymbol{\pi}_j^k$ will imply that file j is requested by cell k and that there will not be any cell t satisfying $r_{t,j} < r_{k,j}$. Considering the definition of $\mathcal{T}_{k,j}$, we obtain that $P_r(\boldsymbol{\pi}_j^k) = (1-\alpha_{k,j})\prod_{t\in\mathcal{T}_{k,j}} \alpha_{t,j}$. Summing up all types of user request profiles $\{\boldsymbol{\pi}_j^k\}$ for all files, the average backhaul rate can be written as (5) which ends the proof of the lemma.

Appendix B

Here pairwise comparison is used to tackle the problem caused by the uncertain relation of $\{\alpha_{\vartheta_\nu,j}\}$. Firstly, we utilize a simple example to help better clarify this lemma.

Example 1: Let $K = 3$. Then it follows that $\boldsymbol{\alpha}^j = [\alpha_{1,j}, \alpha_{2,j}, \alpha_{3,j}]$. Now, assume that for any given j, the only three nonzero elements of $\{y_{k,j}^t\}$ are given by $y_{1,j}^{\theta_1} = 1$, $y_{2,j}^{\theta_2} = 1$, $y_{3,j}^{\theta_3} = 1$. Then we let $\varphi_{t,j} = \left(1-\alpha_{\theta_t,j}\right)\prod_{\nu=1}^{t-1}(\alpha_{\theta_\nu,j})$, $\forall t$ using (12). As such, the objective function can be rewritten as

$$R_{\text{multicast}}^{\text{MDS}} = \left(1-g_{1,j}\right)\left(1-\alpha_{\theta_1,j}\right) + \left(1-g_{2,j}\right)\left(1-\alpha_{\theta_2,j}\right) \times \alpha_{\theta_1,j} + \left(1-g_{3,j}\right)\left(1-\alpha_{\theta_3,j}\right)\alpha_{\theta_1,j}\alpha_{\theta_2,j}. \quad (40)$$

Now we prove that the optimal $\{y_{k,j}^t\}$ must ensure that $\alpha_{\theta_1,j} \geq \alpha_{\theta_2,j} \geq \alpha_{\theta_3,j}$ by contradiction. Assume $\alpha_{\theta_2,j} < \alpha_{\theta_3,j}$ and calculate $R_{\text{multicast}}^{\text{MDS}}$ using (40). Then we exchange the values of $\alpha_{\theta_2,j}$ and $\alpha_{\theta_3,j}$ and recalculate the objective function. The difference between the former and the later objective function can be given by

$$\Delta R_{\text{multicast}}^{\text{MDS}} = \left(g_{3,j} - g_{2,j}\right)\alpha_{\theta_1,j}\left(\alpha_{\theta_3,j} - \alpha_{\theta_2,j}\right). \quad (41)$$

Considering $g_{2,j} \leq g_{3,j}$ and $\alpha_{\theta_2,j} < \alpha_{\theta_3,j}$, we prove that $\Delta R_{\text{multicast}}^{\text{MDS}} \geq 0$. That is to say, for any $\alpha_{\theta_2,j} < \alpha_{\theta_3,j}$, we can always obtain a smaller or at least equal objective function by exchanging $\alpha_{\theta_2,j}$ and $\alpha_{\theta_3,j}$. Hence, $\alpha_{\theta_2,j} \geq \alpha_{\theta_3,j}$ is essential to minimize the backhaul load. In the same way, we can prove that $\alpha_{\theta_1,j} \geq \alpha_{\theta_2,j}$. Consequently, $\alpha_{\theta_1,j} \geq \alpha_{\theta_2,j} \geq \alpha_{\theta_3,j}$ is proved. The same conclusion can easily be extended to the K cell scenario which indicates that $\alpha_{\theta_1,j} \geq \alpha_{\theta_2,j} \geq \cdots \alpha_{\theta_K,j}$. The rigorous mathematical proof is presented below.

We let $\phi_\nu^j, \forall \nu = 2, 3, \ldots, K$ be the summation of the items in $R_{\text{multicast}}^{\text{MDS}}$ that involves $\alpha_{\vartheta_{\nu-1},j}$ and $\alpha_{\vartheta_\nu,j}$, given by

$$\phi_\nu^j = \sum_{k=\nu-1}^{\nu}\left(1-g_{k,j}\right)\left(1-\alpha_{\vartheta_k,j}\right)\prod_{t=1}^{k-1}\alpha_{\vartheta_t,j}s_j. \quad (42)$$

Since $\alpha_{\vartheta_t,j}, t = 1, 2, \ldots, \nu-2$ are interchangeable in ϕ_ν^j, the relation among them will not affect the value of ϕ_ν^j as well as the relation between $\alpha_{\vartheta_{\nu-1},j}$ and $\alpha_{\vartheta_\nu,j}$. Consequently, we consider the derivatives of $\alpha_{\vartheta_{\nu-1},j}$ and $\alpha_{\vartheta_\nu,j}$ in ϕ_ν^j as follows

$$\frac{\partial\phi_\nu^j}{\partial\alpha_{\vartheta_{\nu-1},j}} = \left(g_{\nu-1,j} - 1 + \left(1-g_{\nu,j}\right)\left(1-\alpha_{\vartheta_\nu,j}\right)\right)\prod_{t=1}^{\nu-2}\alpha_{\vartheta_t,j}s_j, \quad (43)$$

$$\frac{\partial\phi_\nu^j}{\partial\alpha_{\vartheta_\nu,j}} = -\left(1-g_{\nu,j}\right)\prod_{t=1}^{\nu-1}\alpha_{\vartheta_t,j}s_j. \quad (44)$$

Let $\Delta_\nu^j = \frac{\partial\phi_\nu^j}{\partial\alpha_{\vartheta_\nu,j}} - \frac{\partial\phi_\nu^j}{\partial\alpha_{\vartheta_{\nu-1},j}}$, and we obtain that

$$\Delta_\nu^j = \sum_{j=1}^{N}\left(1-g_{\nu,j}\right)\left(\alpha_{\vartheta_\nu,j} - \alpha_{\vartheta_{\nu-1},j}\right)\prod_{t=1}^{\nu-2}\alpha_{\vartheta_t,j}s_j + \left(g_{\nu,j} - g_{\nu-1,j}\right)\prod_{t=1}^{\nu-2}\alpha_{\vartheta_t,j}s_j. \quad (45)$$

Because $\Delta_\nu^j = 0$ indicates that $\alpha_{\vartheta_{\nu-1},j}$ and $\alpha_{\vartheta_\nu,j}$ are interchangeable, here we focus on the case when $\Delta_\nu^j \neq 0$. If $\Delta_\nu^j > 0$, it follows that the derivative of $\alpha_{\vartheta_\nu,j}$ in ϕ_ν^j is higher than that of $\alpha_{\vartheta_{\nu-1},j}$, which is to say, the weight for $\alpha_{\vartheta_\nu,j}$ in terms of the weighted summation ϕ_ν^j is higher. Hence, we should let $\alpha_{\vartheta_\nu,j} \leq \alpha_{\vartheta_{\nu-1},j}$ in order to minimize the objective function $R_{\text{multicast}}^{\text{MDS}}$. On the contrary, if $\Delta_\nu^j < 0$, then it holds true that $\alpha_{\vartheta_\nu,j} \geq \alpha_{\vartheta_{\nu-1},j}$. Consequently, assuming that $\Delta_\nu^j < 0$, we obtain $\alpha_{\vartheta_\nu,j} \geq \alpha_{\vartheta_{\nu-1},j}$ and hence the right side of (45) is always non-negative since $g_{\nu,j} \leq 1$ and $g_{\nu-1,j} \leq g_{\nu,j}$, which conflicts with the assumption. Hence, it holds true that $\Delta_\nu^j \geq 0$ and $\alpha_{\vartheta_\nu,j} \leq \alpha_{\vartheta_{\nu-1},j}$ and the lemma is then proved.

Based on the definition of $\boldsymbol{\beta}^j$ and the conclusion drawn above, we derive that $\boldsymbol{\beta}^j = [\alpha_{\theta_1,j}, \alpha_{\theta_2,j}, \ldots, \alpha_{\theta_K,j}]$. As a

consequence, the optimal $\varphi^*_{t,j}$ can be written as $\varphi^*_{t,j} = \left(1-\beta_{t,j}\right)\prod_{\nu=1}^{t-1}\beta_{\nu,j}$. The corresponding values of $\{y^t_{k,j}\}$ can easily be calculated as given in (18).

Appendix C

Given some cooperative caching policy $(\{x^t_{k,j}\},\{m_{k,j}\})$, the costs for fetching content from neighboring cells are the same in the coded and uncoded caching scenarios. Therefore, the difference in the backhaul cost shows up most clearly in the UA costs. When uncoded fragments are stored, all the fragments except the ones that are either stored in local cache or fetched from the neighboring cells are needed from the MBS via backhual to each cell requesting the particular file. Considering the possible content overlap amongst those fragments, the number of unique fragments for file j available at cell $k \in \mathcal{K}_{\boldsymbol{\pi}_j}$ would always be less than or equal to $\sum_t x^t_{k,j}$ for a certain user request profile $\boldsymbol{\pi}_j$ which leads to a higher backhaul rate than that in the MDS coded case. If the fragments are assumed to be randomly selected to be stored in the cells and then sent to the neighboring cells *equiprobably*, the probability of each fragment of file j needing to be sent to cell k via backhaul, i.e., not being stored locally or sent to the particular cell k from other SBSs, would be given by

$$\widehat{\rho}_{k,j} = \prod_{t=1}^{K}\left(\frac{\binom{n_j-1}{m_{t,j}}}{\binom{n_j}{m_{t,j}}} + \frac{\binom{n_j-1}{m_{t,j}-1}}{\binom{n_j}{m_{t,j}}}\frac{\binom{m_{t,j}-1}{x^t_{k,j}}}{\binom{m_{t,j}}{x^t_{k,j}}}\right) = \prod_{t=1}^{K}\left(1-\frac{x^t_{k,j}}{n_j}\right). \tag{46}$$

In this case, the average UA cost can be written as

$$C^{\text{uncoded}}_{\text{coop}} = \sum_{j=1}^{N}\sum_{k=1}^{K}\left[\widehat{\rho}_{k,j}\, f^M_k + \sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j} f^t_k\right] s_j(1-\alpha_{k,j}). \tag{47}$$

Compared with the UA cost in (27), if we can prove that

$$\prod_{t=1}^{K}\left(1-\frac{x^t_{k,j}}{n_j}\right) \geq 1-\min\left(1,\sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j}\right), \quad \forall k,j, \tag{48}$$

then it holds true that $C^{\text{MDS}}_{\text{coop}} \leq C^{\text{uncoded}}_{\text{coop}}$. Hence, here we focus on the proof of the result (48). As can be observed, when $\sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j} \geq 1$, (48) is always true. When $\sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j} < 1$, the right hand side of (48) equals to $\left(1-\sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j}\right)$. In this case, we prove (48) using mathematical induction.

To be brief, we mathematically reformulate the problem into a general problem, which reads

$$\prod_{t=1}^{K}(1-\chi_t) \geq 1-\sum_{t=1}^{K}\chi_t, \tag{49}$$

where $\chi_t \in [0,1]$. Obviously, when $K=1$ or 2, the statement is always true as expected. Now assuming that (49) holds for $K=\kappa$, we hence have

$$\prod_{t=1}^{\kappa}(1-\chi_t) \geq 1-\sum_{t=1}^{\kappa}\chi_t. \tag{50}$$

Then it follows that

$$\begin{aligned}\prod_{t=1}^{\kappa+1}(1-\chi_t) &= \prod_{t=1}^{\kappa}(1-\chi_t) - \prod_{t=1}^{\kappa}(1-\chi_t)\,\chi_{\kappa+1} \\ &\geq \left(1-\sum_{t=1}^{\kappa}\chi_t\right) - \chi_{\kappa+1},\end{aligned} \tag{51}$$

due to the fact that $0 \leq \prod_{t=1}^{\kappa}(1-\chi_t) \leq 1$ as well as the inequality (50). Now we are able to conclude that the statement is true for all available K via induction. Then going back to the original problem and letting $\chi_t = \frac{x^t_{k,j}}{n_j}$ for any given k, we have proved the statement

$$\prod_{t=1}^{K}\left(1-\frac{x^t_{k,j}}{n_j}\right) \geq 1-\sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j}, \quad \forall k,j. \tag{52}$$

Based on this analysis, $C^{\text{MDS}}_{\text{coop}} \leq C^{\text{uncoded}}_{\text{coop}}$ is then proved.

Appendix D

Considering multicast-aware cooperative caching, the UA cost can be written as

$$\begin{aligned}C^{\text{Mul}}_{\text{coop}} = \sum_{j=1}^{N}\sum_{\boldsymbol{\pi}_j\in\Pi_j}\Bigg[&\left(1-\min_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}}\sum_{t=1}^{K}z^t_{k,j}\right)\max_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} f^M_k \\ &+ \sum_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}}\sum_{t=1}^{K}z^t_{k,j}\, f^t_k\Bigg] P_r(\boldsymbol{\pi}_j)s_j.\end{aligned} \tag{53}$$

As we can see, the first item denotes the backhaul cost while the second item presents the cost for content sharing among the cooperative SBSs. For each given user request profile for a particular file $\boldsymbol{\pi}_j$, the cost for fetching content from the cooperative SBSs at cell k appears only when file j is requested by the users in cell k which means that $\boldsymbol{\pi}_j(k)=1$ regardless of the individual user request profiles in other cells. It is easy to prove $P_r(\boldsymbol{\pi}_j|_{\boldsymbol{\pi}_j(k)=1}) = 1-\alpha_{k,j}$, and so (28).

Appendix E

If $(\{x^t_{k,j}\},\{m_{k,j}\})$ is given, then the costs for fetching content from neighboring cells will be the same in the coded and uncoded caching scenarios. As a result, the comparison is focused on the backhaul costs in the two scenarios. When uncoded fragments are stored, all the fragments except for the ones that can be fetched at all of the cells requesting the file either from local cache or from the neighboring cells are needed to be sent from the MBS via multicast transmission. Assuming that the fragments are randomly selected to be stored in the cells and then sent to the neighboring cells *equiprobably*, the probability of each fragment of file j available at all of the cells requesting the file either from local cache or from the neighboring cells would be given by

$$\widetilde{\rho}_{\boldsymbol{\pi}_j} = \prod_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}}(1-\widehat{\rho}_{k,j}), \tag{54}$$

where $\widehat{\rho}_{k,j}$ is the probability of each fragment of file j not being stored locally or sent to the particular cell k from other

SBSs given by (46) in Appendix A. Similar to the multicast-aware case, the average UA cost can be written as

$$C_{\text{mult,coop}}^{\text{uncoded}} = \sum_{j=1}^{N}\Bigg[\sum_{\boldsymbol{\pi}_j\in\Pi_j}\left(1-\widetilde{\rho}_{\boldsymbol{\pi}_j}\right)\max_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} f_k^M P_r(\boldsymbol{\pi}_j) + \sum_{k=1}^{K}\sum_{t=1}^{K} z_{k,j}^t f_k^t(1-\alpha_{k,j})\Bigg] s_j. \quad (55)$$

According to (46) and (52), we obtain

$$\widetilde{\rho}_{\boldsymbol{\pi}_j} \le \prod_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} \left(\sum_{t=1}^{K}\frac{x_{k,j}^t}{n_j}\right). \quad (56)$$

As $0 \le \sum_{t=1}^{K}\frac{x_{k,j}^t}{n_j} \le 1, \forall k \in \mathcal{K}_{\boldsymbol{\pi}_j}$, it holds true that $\widetilde{\rho}_{\boldsymbol{\pi}_j} \le \min_{k\in\mathcal{K}_{\boldsymbol{\pi}_j}} \sum_{t=1}^{K} z_{k,j}^t$. Compared with the average UA cost in (28), we derive that $C_{\text{mult,coop}}^{\text{MDS}} \le C_{\text{mult,coop}}^{\text{uncoded}}$.

APPENDIX F

To proceed, we sort $\boldsymbol{\lambda}^j = \{\lambda_{k,j}, k \in \mathcal{S}_l^u\}$ in an ascending order and define the sorted vector as $\boldsymbol{\psi}^j$ with $\psi_{k,j} = \lambda_{\vartheta_k,j}$ and $\psi_{k,j} \le \psi_{k+1,j}, \forall k \in \mathcal{S}_l^u \setminus |\mathcal{S}_l^u|$. For instance, if $\vartheta_1 = k$, it means that $\lambda_{k,j}$ equals to $\psi_{1,j}$ and is therefore the lowest. On the contrary, if $\vartheta_{|\mathcal{S}_l^u|} = k$, it means that $\lambda_{k,j}$ equals to $\psi_{|\mathcal{S}_l^u|,j}$ and is hence the highest. Consequently, the objective function in (36) can be rewritten as

$$\widetilde{C}_l^u = \sum_{j=1}^{N}\sum_{k\in\mathcal{S}_l^u}\Bigg[\left(1-\psi_{k,j}\right) f_u^M(1-\alpha_{\vartheta_k,j})\prod_{\nu=1}^{k-1}\alpha_{\vartheta_\nu,j} + \psi_{k,j} f_l^u(1-\alpha_{\vartheta_k,j})\Bigg] s_j. \quad (57)$$

The reformulated problem can then be written as

$$\min_{\{\psi_{k,j}\}} \ \widetilde{C}_l^u \quad (58a)$$

$$\text{s.t. } \psi_{k,j} \le \psi_{k+1,j}, \quad \forall k \in \mathcal{S}_l^u \setminus |\mathcal{S}_l^u|, \ \forall j, \quad (58b)$$

$$0 \le \psi_{k,j} \le 1, \quad \forall k \in \mathcal{S}_l^u, \ \forall j, \quad (58c)$$

$$q_{l,j}^u \le \psi_{k,j} \le |\mathcal{S}_l^u| q_{l,j}^u, \quad \forall k \in \mathcal{S}_l^u, \ \forall j. \quad (58d)$$

Apparently, $\psi_{k,j}, \forall k \in \mathcal{S}_l^u$ are treated similarly in the constraints (58c)-(58d) regardless of the values of $\{\vartheta_k\}$. Given any $\{\vartheta_k\}$, we want to find the actual relation of the optimal $\psi_{k,j}, \forall k \in \mathcal{S}_l^u$ to minimize the objective function in (57). Furthermore, the objective function and constraints are independent towards of different files in (58), and hence the UA cost minimization problem for each cluster can be further decomposed into N sub-problems each minimizing the associated cost for a particular file defined as $\widetilde{C}_{l,j}^u, \forall j$. Thus, we consider the derivatives of $\{\psi_{k,j}\}$ in $\widetilde{C}_{l,j}^u$ given by

$$\frac{\partial \widetilde{C}_{l,j}^u}{\partial \psi_{k,j}} = \left(Q_l^u - \prod_{\nu=1}^{k-1}\alpha_{\vartheta_\nu,j}\right) f_u^M(1-\alpha_{\vartheta_k,j}) s_j, \forall k \in \mathcal{S}_l^u \setminus 1, \quad (59)$$

$$\frac{\partial \widetilde{C}_{l,j}^u}{\partial \psi_{1,j}} = (Q_l^u - 1) f_u^M(1-\alpha_{\vartheta_k,j}) s_j, \quad (60)$$

where $Q_l^u = f_l^u / f_u^M$ denotes the ratio between the costs of fetching content via backhaul and from the cluster. Since $0 < Q_l^u < 1$, it holds true that $\frac{\partial \widetilde{C}_{l,j}^u}{\partial \psi_{1,j}} < 0$. For any $\psi_{k,j}, \forall k \in \mathcal{S}_l^u \setminus 1$ satisfying the constraints, $\widetilde{C}_{l,j}^u$ reaches its lowest when we let $\psi_{1,j} = \psi_{2,j}$ since a larger $\psi_{1,j}$ contributes to a lower $\widetilde{C}_{l,j}^u$. In the same way, it can be proved that the relation between $\psi_{k,j}$ and $\psi_{k+1,j}$ is subject to the value of $\left(Q_l^u - \prod_{\nu=1}^{k-1}\alpha_{\vartheta_\nu,j}\right)$. Note that $\prod_{\nu=1}^{k-1}\alpha_{\vartheta_\nu,j}$ always decreases with the increase of k which indicates that if $\prod_{\nu=1}^{k-1}\alpha_{\vartheta_\nu,j} \le Q_l^u$, we always have $\prod_{\nu=1}^{t-1}\alpha_{\vartheta_\nu,j} < Q_l^u, \forall t > k$. Hence, we discuss about the relation among $\{\psi_{k,j}\}$ in two kinds of conditions. In the first case, we assume that $Q_l^u \le \prod_{\nu=1}^{|\mathcal{S}_l^u|-1}\alpha_{\vartheta_\nu,j}$, and it is easy to prove that $\psi_{1,j} = \psi_{2,j} = \cdots = \psi_{|\mathcal{S}_l^u|,j}$ by iteratively utilizing the similar trick for proving $\psi_{1,j} = \psi_{2,j}$. Otherwise, when

$$Q_l^u \ge \prod_{\nu=1}^{k-1}\alpha_{\vartheta_\nu,j} = \begin{cases} < 0, & k \in [1,\ldots,t], \\ \ge 0, & k \in [t+1,\ldots,|\mathcal{S}_l^u|], \end{cases} \quad (61)$$

it is still possible to prove that $\psi_{1,j} = \psi_{2,j} = \cdots = \psi_{t+1,j}$ by fixing $\psi_{t+1,j}$. While for $k \in [t+1,\ldots,|\mathcal{S}_l^u|]$ when $\widetilde{C}_{l,j}^u$ decreases with the decline of $\psi_{k,j}$, we let $\psi_{t+1,j} = \psi_{t+2,j} = \cdots = \psi_{|\mathcal{S}_l^u|,j}$ to get the lowest cost $\widetilde{C}_{l,j}^u$ for any given $\psi_{t+1,j}$ using (58b). It is then proved that $\psi_{1,j} = \psi_{2,j} = \cdots = \psi_{|\mathcal{S}_l^u|,j}$. As a result, we derive that $\lambda_{k,j} = \lambda_{t,j}, \forall k, t \in \mathcal{S}_l^u$.

REFERENCES

[1] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 16–22, Aug. 2016.

[2] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Jun. 2015, pp. 3358–3363.

[3] V. Sourlas, P. Georgatsos, P. Flegkas, and L. Tassiulas, "Partition-based caching in information-centric networks," in *Proc. IEEE Int. Workshop Netw. Sci. Commun. Netw.*, Hong Kong, Apr. 2015, pp. 396–401.

[4] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[5] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[6] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," in *Proc. IEEE Inf. Theory Appl. Workshop (ITA)*, San Diego, CA, USA, Feb. 2015, pp. 98–107.

[7] J. Zhang, X. Lin, C.-C. Wang, and X. Wang, "Coded caching for files with distinct file sizes," in *Proc. IEEE Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1686–1690.

[8] S. Wang, W. Li, X. Tian, and H. Liu. (Apr. 2016). "Fundamental limits of heterogenous cache." [Online]. Available: https://arxiv.1504.01123v1

[9] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "On the average performance of caching and coded multicasting with random demands," in *Proc. IEEE Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2014, pp. 922–926.

[10] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 355–370, Feb. 2015.

[11] F. Xu, K. Liu, and M. Tao, "Cooperative Tx/Rx caching in interference channels: A storage-latency tradeoff study," in *Proc. IEEE Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 2034–2038.

[12] 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.

[13] R. Tandon and O. Simeone, "Cloud-aided wireless networks with edge caching: Fundamental latency trade-offs in fog radio access networks," in *Proc. IEEE Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 2029–2033.

[14] 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, Sep. 2016.

[15] X. Peng, J. C. Shen, J. Zhang, and K. B. Letaief, "Joint data assignment and beamforming for backhaul limited caching networks," in *Proc. IEEE Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Washington, DC, USA, Sep. 2014, pp. 1370–1374.

[16] A. Liu and V. K. N. Lau, "Mixed-timescale precoding and cache control in cached MIMO interference network," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6320–6332, Dec. 2013.

[17] E. Baştuğ, M. Bennis, M. Kountouris, and M. Debbah, "Cache-enabled small cell networks: Modeling and tradeoffs," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, p. 41, Feb. 2015.

[18] 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.

[19] Z. Chen, J. Lee, T. Q. Quek, and M. Kountouris. (Jan. 2016). "Cooperative caching and transmission design in cluster-centric small cell networks." [Online]. Available: https://arxiv.org/abs/1601.00321

[20] S. Tamoor-ul-Hassan, M. Bennis, P. H. J. Nardelli, and M. Latva-Aho, "Caching in wireless small cell networks: A storage-bandwidth tradeoff," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1175–1178, Jun. 2016.

[21] K. Poularakis and L. Tassiulas, "On the complexity of optimal content placement in hierarchical caching networks," *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 2092–2103, Mar. 2016.

[22] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 77–83, Aug. 2016.

[23] Y. Cui, F. Lai, S. Hanly, and P. Whiting, "Optimal caching and user association in cache-enabled heterogeneous wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[24] E. Altman, K. Avrachenkov, and J. Goseling. (Sep. 2013). "Coding for caches in the plane." [Online]. Available: https://arxiv.org/abs/1309.0604

[25] D. Jiang and Y. Cui. (Oct. 2016). "Partition-based caching in large-scale SIC-enabled wireless networks." [Online]. Available: https://arxiv.org/abs/1610.09526

[26] V. Bioglio, F. Gabry, and I. Land, "Optimizing MDS codes for caching at the edge," in *Proc. IEEE GLOBECOM*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[27] N. Abedini and S. Shakkottai, "Content caching and scheduling in wireless networks with elastic and inelastic traffic," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 864–874, Jun. 2014.

[28] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting caching and multicast for 5G wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2995–3007, Jan. 2016.

[29] J. Liao, K. K. Wong, M. R. A. Khandaker, and Z. Zheng, "Optimizing cache placement for heterogeneous small cell networks," *IEEE Commun. Lett.*, vol. 21, no. 1, pp. 120–123, Sep. 2016.

[30] Y. Cui, D. Jiang, and Y. Wu, "Analysis and optimization of caching and multicasting in large-scale cache-enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 5101–5112, Jul. 2016.

[31] Y. Cui and D. Jiang, "Analysis and optimization of caching and multicasting in large-scale cache-enabled heterogeneous wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 250–264, Jan. 2017.

[32] A. Gharaibeh, A. Khreishah, B. Ji, and M. Ayyash, "A provably efficient online collaborative caching algorithm for multicell-coordinated systems," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 1863–1876, Aug. 2016.

[33] 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, Aug. 2016.

[34] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "In-network caching and content placement in cooperative small cell networks," in *Proc. 5GU*, Levi, Finland, Nov. 2014, pp. 128–133.

[35] X. Peng, J. Zhang, S. H. Song, and K. B. Letaief, "Cache size allocation in backhaul limited wireless networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[36] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Design and evaluation of the optimal cache allocation for content-centric networking," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 95–107, Jan. 2016.

[37] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Trans. Commun.*, vol. 62, no. 10, pp. 3665–3677, Oct. 2014.

[38] D. Lopez-Perez, I. Guvenc, G. de la Roche, M. Kountouris, T. Q. S. Quek, and J. Zhang, "Enhanced intercell interference coordination challenges in heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 22–30, Jun. 2011.

[39] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, New York, NY, USA, Mar. 1999, pp. 126–134.

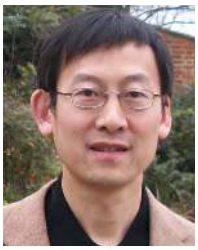
[40] M. Grant and S. Boyd. (Sep. 2013). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.0 Beta*. [Online]. Available: http://cvxr.com/cvx

Jialing Liao (S'17) received the B.S. and M.S. degrees in underwater acoustic engineering from Harbin Engineering University, Harbin, China, in 2012 and 2014, respectively. She is currently pursuing the Ph.D. degree in electronic engineering from University College London, London, U.K. Her research interests lie in the field of network optimization with emphasis on wireless edge caching for content centric networks.

Kai-Kit Wong (SM'08–F'16) received the B.Eng., M.Phil., and Ph.D. degrees from The Hong Kong University of Science and Technology, Hong Kong, in 1996, 1998, and 2001, respectively, all in electrical and electronic engineering. He took up faculty and visiting positions at The University of Hong Kong, Lucent Technologies, Bell-Labs, Holmdel, NJ, USA, the Smart Antennas Research Group, Stanford University, and the Department of Engineering, University of Hull, U.K. He is currently a Professor of wireless communications with the Department of Electronic and Electrical Engineering, University College London, U.K. He is a fellow of the IET. He also served as an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2005 to 2011 and the IEEE SIGNAL PROCESSING LETTERS from 2009 to 2012. He serves on the Editorial Board of the IEEE WIRELESS COMMUNICATIONS LETTERS, the IEEE COMMUNICATIONS LETTERS, and the IEEE COMSOC/KICS JOURNAL OF COMMUNICATIONS AND NETWORKS.

Yangyang Zhang received the B.S. and M.S. degrees in electronics and information engineering from Northeastern University, Shenyang, China, in 2002 and 2004, respectively, and the Ph.D. degree in electrical engineering from the University of Oxford, in 2008. From 2008 to 2010, he was a Post-Doctoral Research Fellow with the University College London. He is currently with the Shenzhen Key Laboratory of Artificial Microstructure Design and Guangdong Key Laboratory of Meta-RF Microwave Radio Frequency, Kuang-Chi Institute of Advanced Technology, Shenzhen, China, where he is also the Executive Vice President. He has published around 40 papers in various journals and conferences. His research interests are mainly focused on metamaterial-based future wireless communication system, such as MIMO communication system, metamaterial-based RF devices, and metamaterial-based spatial modulation technology. He received over 20 honors from various national and international competitions.

Zhongbin Zheng received the bachelor's and master's degrees in information and communications engineering from the Beijing University of Posts and Telecommunications in 2002 and 2005, respectively. He is also the former Head of the Technology Department, East China Institute, Ministry of Industry and Information Technology. He is currently the Vice Director of the East China Institute of Telecommunications, China Academy of Information and Communications Technology. He is very active in research, resulting in not only a number of international paper publications, but also patents and draft standards.



Kun Yang (SM'08) received the B.Sc. and M.Sc. degrees from the Computer Science Department, Jilin University, China, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, University College London (UCL), U.K. He was with UCL on several European Union (EU) research projects for several years. He is currently a Chair Professor with the School of Computer Science and Electronic Engineering, University of Essex, leading the Network Convergence Laboratory, U.K. He is also an affiliated Professor with UESTC, China. His main research interests include wireless networks and communications, future Internet technology and network virtualization, and mobile cloud computing. He manages research projects funded by various sources, such as U.K. EPSRC, EU FP7/H2020, and industries. He has authored over 100 journal papers. He serves on the editorial boards of both IEEE and non-IEEE journals. He has been a fellow of the IET since 2009.