Collaborative Hierarchical Caching for Traffic Offloading in Heterogeneous Networks

Xiuhua Li[1], Xiaofei Wang[2], Keqiu Li [2], and Victor C. M. Leung[1]
[1]Dept. Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada
[2]Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology,
Tianjin University, Tianjin, China.
Email: {lixiuhua, vleung}@ece.ubc.ca, xiaofeiwang@tju.edu.cn, likeqiu@gmail.com

***Abstract*—To address the challenge arising from mobile users' increasing demands for multimedia services, applying content caching in heterogeneous networks (HetNets) is regarded as an effective way to offload traffic and improve the capacity of mobile networks. In this paper, we aim at designing novel content caching strategies in HetNets to offload the network traffic and support users' requests locally. Specifically, based on some practical network constraints (i.e., patterns of user requests, link capacity and heterogeneous cache sizes) and the derived network topology, we propose a low-complexity and practicable distributed collaborative hierarchical caching framework by decomposing the formulated large-scale optimization problem into a series of simpler subproblems. Trace-based simulation results demonstrate the effectiveness of the proposed framework.**

I. Introduction

Current mobile networks need to download multimedia contents directly from service providers (SPs) over the Internet to satisfy mobile users' service requests. However, the resulting mobile network traffic is expected to increase by an order of magnitude over the next five years due to the explosive growth in multimedia content requests, which has become one of most important concerns of mobile network operators (MNOs) due to the scarcity of resources, especially in the radio access networks and backhaul networks [1]. It is necessary to address this challenge by deploying advanced networking techniques over new network architectures being developed for next generation (i.e., 5G) mobile systems [2]–[5].

One emerging technique is to deploy caches at the edges of mobile networks to bring contents closer to mobile users [1]–[5]. Since the popularity of contents has been found to follow the "Power Law" [6], caching popular contents inside mobile networks can significantly reduce the massive duplicated content downloads and data transmissions from SPs outside the MNO network, thereby offloading network traffic effectively and supporting users' content requests locally [5]. Moreover, the heterogeneous network architecture is effective in enhancing wireless link quality between mobile users and base stations (BSs) and thus improving network capacity [7]. Generally, heterogeneous nodes such as macro BSs, micro BSs, pico BSs, femto BSs and relays are deployed in a HetNet, which forms a multi-tier network architecture. However, to support users' content requests, high-capacity backhauls are needed between different types of BSs/relays and SPs to accommodate the resulting increase in network traffic. Considering the benefits of content caching and HetNets, it is attractive to combine them in cache-enabled HetNets.

Considering the scale of popular contents and the scarcity of network resources, it is important to design caching schemes that utilize the network infrastructures effectively. In particular, many factors need to be considered when caching a content, e.g., its popularity, storage size, locations of existing replicas in the network topology and so on [4], [5]. There are serval studies on content caching at BSs in mobile networks. For instance, collaborative multi-cell caching in [5] and FemtoCaching in [8] were proposed to cache popular contents at BSs in small cells to offload network traffic and increase the number of served users. Collaborative BS caching schemes in [4], [9], [10] were proposed to improve users' Quality of Service (QoS) especially on content access delay. Multicast beamforming schemes in [11], [12] were proposed for content delivery from BSs to users through wireless links. However, these studies only focus on the case of single-tier caching and do not utilize hierarchical network infrastructures of HetNets effectively. There exist only few studies on hierarchical caching in HetNets. The theoretical performance analysis in [7] on content caching in HetNets did not address the design of caching schemes. A collaborative hierarchical caching framework in HetNets was proposed in [13] to improve users' QoS. Both of these works assumed content sizes to be identical, which is not practical in practical systems. The design of collaborative hierarchical caching in HetNets with practical considerations is still not well explored, especially on offloading network traffic to reduce the MNO's operational expenditure.

The gap in the literature as discussed above motivates us to focus on designing a novel collaborative hierarchical content caching framework in HetNets to offload the network traffic and support users' requests locally. The problem is formulated as a minimization of the total network traffic load of accessing the requested contents under practical network constraints, i.e., patterns of user requests, link capacity and heterogeneous cache sizes. By decomposing the formulated large-scale optimization problem into a series of simpler subproblems based on the derived hierarchical caching topology, we propose a low-complexity distributed collaborative hierarchical caching

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

framework and a content request routing scheme for practical implementation. Simulation results based on real-world traces demonstrate the effectiveness of the proposed framework.

The remainder of this paper is organized as follows. We describe the system model and formulate the hierarchical caching optimization problem in Section II. In Section III, we propose a low-complexity collaborative hierarchical caching framework. Trace-based simulation results are shown in Section IV to evaluate the proposed caching framework. Finally, Section V concludes the paper.

II. System Model and Problem Formulation

A. System Model

Fig. 1 [5] illustrates the hierarchical caching architecture in a HetNet. Multimedia contents are offered over the Internet by SPs (e.g., YouTube, Facebook) outside the MNO network. Meanwhile, inside the MNO network, a great number of macro cells cover the whole service area to satisfy the services of mobile users. In each macro cell, several femto cells cover most of the local macro cell's area. Limited-capacity backhaul links (e.g., cables or optical fibers) connect each macro BS (MBS) with its local geographically distributed femto BSs (FBSs), and some FBSs to each other. Mobile users are divided into macro users and femto users that are associated with MBSs and FBSs, respectively. Content requests from users are received and served by their associated BSs.

Moreover, to satisfy users' content requests and reduce the network traffic load from duplicated content delivery, both MBSs and FBSs are able to cache some contents with limited heterogeneous cache storage capacity, which forms a two-tier caching topology as shown in Fig. 2. Here, the tiers of MBSs and FBSs in the HetNet are defined as Tier 1 and Tier 2, respectively. Besides, each MBS decides how to effectively select popular contents to cache in its connected FBSs and its own caching storage, and maintains a list of all the BSs' cached contents in the macro cell at the cost of a small signaling overhead that is assumed to be negligible. Moreover, we assume that the popularity of contents changes slowly, and can be determined in advance or predicted by the system through learning and analysing users' behavior and preference. Thus, the information of content popularity is available in the network.

In the hierarchical caching topology in Fig. 2, to satisfy a dynamic content request from a user in a cell, the associated local BS either returns the content if it is locally available, or routes the request to other BSs or downloads the content directly from the Internet via the MNO core. The details of a content request routing scheme will be proposed in the following section. From the perspective of the network, it is essential to satisfy users' content requests while offloading the network traffic as much as possible.

In this paper, we consider the case of a single macro cell, equipped with a MBS and N FBSs (denoted by $\{\mathrm{FBS}_1, \mathrm{FBS}_2, \ldots, \mathrm{FBS}_N\}$) to serve U active users. The storage capacities of the FBSs' caches are denoted by $\{S_1, S_2, \ldots, S_N\}$ while the storage capacity of the MBS is denoted by S_0. There is a catalog of F popular contents, denoted by $\{o_1, o_2, \ldots, o_F\}$. We define three sets as $\mathcal{N} = \{1, 2, \ldots, N\}$, $\mathcal{F} = \{1, 2, \ldots, F\}$ and $\mathcal{U} = \{1, 2, \ldots, U\}$. In practice, contents have various storage sizes denoted by $\{s_1, s_2, \ldots, s_F\}$. The sets of users associated with the MBS and FBSs are denoted by $\{\mathcal{U}_0, \mathcal{U}_1, \ldots, \mathcal{U}_N\}$. Besides, we consider that a user in each cell is only associated with a BS. As a result, we have $\mathcal{U}_0 \cup \mathcal{U}_1 \cup \ldots \cup \mathcal{U}_N = \mathcal{U}$, and $\mathcal{U}_i \cap \mathcal{U}_j = \emptyset$ for $\forall i \neq j \in \mathcal{N} \cup \{0\}$.

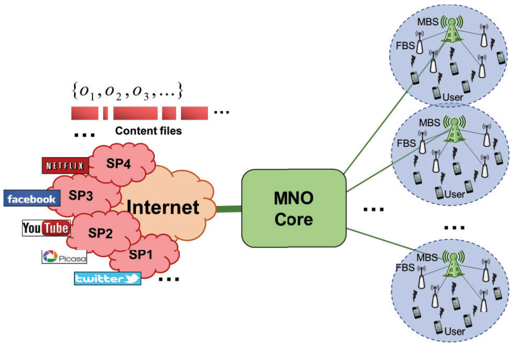

Fig. 1. Illustration of hierarchical caching architecture in HetNets.

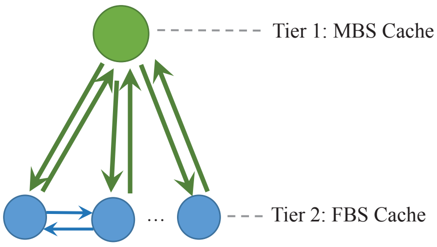

Fig. 2. Topology of hierarchical caching in HetNets.

Moreover, we use the model of user association in multi-tier HetNets as in [14]. Specifically, the considered two tiers of BSs may differ in their spatial density, transmit power and supported data rate. The locations of BSs at Tier i $(i = 1, 2)$ follow a homogeneous Poison point process (PPP) with intensity θ_i. The two tiers have identical path loss exponent γ. Each BS at Tier i has the same transmit power ρ_i and the same signal-to-interference-plus-noise ratio (SINR) threshold Γ_i. The random channel fluctuations are modeled as Rayleigh fading with unit average power, and the interference-limited case neglecting the noise is considered. An open access strategy is used, i.e., a user is allowed to connect to any tier. Besides, a user can be associated with a BS at Tier i only when its SINR w.r.t. the BS is no less than Γ_i. The locations of users are also modeled as a homogeneous PPP that is independent of that of BSs' locations. Thus, based on the analysis in [14], the average proportion of users associated with the BSs at Tier i in the open strategy can be expressed as

$$\alpha_i = \frac{\theta_i \rho_i^{2/\gamma} \Gamma_i^{-2/\gamma}}{\sum_{j=1}^{2} \theta_j \rho_j^{2/\gamma} \Gamma_j^{-2/\gamma}}, \ \forall i \in \{1, 2\}. \tag{1}$$

Specifically, in the considered system, the ratio between the intensity of BSs at the two tiers can be approximately calculated as $\theta_1 : \theta_2 = 1 : N$, and the ratio between the average numbers of users connected to the two tiers is approximately calulated as $|\mathcal{U}_0| : \sum_{n\in\mathcal{N}} |\mathcal{U}_n| = \alpha_1 : \alpha_2$.

Moreover, the overall popularity of the contents in the network, denoted as $\{P_1, P_2, \ldots, P_F\}$, is assumed to satisfy the Mandelbrot-Zipf (MZipf) distribution [15] as

$$P_f = \frac{(\Upsilon_f + \tau)^{-\beta}}{\sum_{i\in\mathcal{F}} (\Upsilon_i + \tau)^{-\beta}},\ \forall f \in \mathcal{F}, \tag{2}$$

where Υ_f is the rank of the content o_f in the descending order of overall content popularity, $\tau \geq 0$ is the plateau factor, and $\beta > 0$ is the skewness factor. Denote the local popularity of content o_f collected at the MBS and FBSs as $\{p_0^f, p_1^f, \ldots, p_N^f\}$. Here, we have $P_f = \sum_{n=0}^{N} p_n^f$. We assume that the probability of accessing content o_f is identical for the users that are associated with the same BS, denoted by $\{q_0^f, q_1^f, \ldots, q_N^f\}$ satisfying $q_n^f = \frac{p_n^f}{\sum_{j\in\mathcal{F}} p_n^j}, \forall n \in \mathcal{N} \cup \{0\}, \forall f \in \mathcal{F}$. Besides, the average arrival rate of content requests from user u is λ_u. Thus, for user $u \in \mathcal{U}_n, n \in \mathcal{N} \cup \{0\}$, the corresponding average arrival rate of requests for accessing content o_f, denoted by λ_{nu}^f, can be calculated as $\lambda_{nu}^f = q_n^f \lambda_u$, and we regard $\lambda_{nu}^f s_f$ as the average bandwidth capacity requirement for the requests for content o_f from user u. The average available link capacity between the MBS and FBS$_n$ is denoted by $C_n, n \in \mathcal{N}$, and the average available link capacity from FBS$_k$ to FBS$_n$ is denoted by $C_{kn}, k \in \mathcal{N}, n \in \mathcal{N}$. If FBS$_k$ is not connected to FBS$_n$, then we set $C_{kn} = 0$.

To reduce the complexity of content management, we assume that any content is either entirely cached or not, and that any content between two BSs is entirely delivered or not. Besides, a femto user is able to access the requested contents from the local FBS or the MBS, while a macro user can only access contents from the MBS. Denote $x_n^f \in \{0,1\}$ for whether FBS$_n$ caches content o_f while $x_0^f \in \{0,1\}$ for whether the MBS caches content o_f, where "1" means caching while "0" means no caching. Besides, denote $y_n^f \in \{0,1\}$ for whether to deliver content o_f from the MBS to FBS$_n$ while $y_{kn}^f \in \{0,1\}$ for whether to deliver content o_f from FBS$_k$ to FBS$_n$ $(n \neq k)$, where "1" means delivery while "0" means no delivery. Besides, we set $y_{nn}^f = 0$ for $\forall n \in \mathcal{N}, \forall f \in \mathcal{F}$. Thus, designing the caching strategies involves finding all the values of $\{x_n^f, x_0^f, y_n^f, y_{kn}^f\}$.

B. Problem Formulation

In this paper, our objective is to minimize the expected sum of traffic load caused by downloading contents directly from the Internet to the MBS via the MNO core, thereby satisfying users' content requests locally in the considered macro cell. Consequently, with caching strategies, the corresponding expected sum of traffic load from satisfying femto users' content requests can be calculated as

$$TF_1 = F \sum_{n\in\mathcal{N}} \sum_{f\in\mathcal{F}} \sum_{u\in\mathcal{U}_n} \big(1 - x_n^f - y_n^f - \sum_{k\in\mathcal{N}} y_{kn}^f\big) \lambda_{nu}^f s_f. \tag{3}$$

The corresponding expected sum of traffic load from satisfying marco users' content requests can be calculated as

$$TF_2 = F \sum_{f\in\mathcal{F}} \sum_{u\in\mathcal{U}_0} \big(1 - x_0^f\big) \lambda_{0u}^f s_f. \tag{4}$$

Thus, within the limits of the cache storage sizes of all the BSs and the capacity between the MBS and FBSs, the overall problem of minimizing the expected sum of traffic load caused by downloading contents directly from the Internet to the MBS via the MNO core in the network can be formulated as

$$\min_{\{x_n^f, x_0^f, y_n^f, y_{kn}^f\}} \quad TF_1 + TF_2 \tag{5a}$$

$$\textit{s.t.} \sum_{f\in\mathcal{F}} x_n^f s_f \leq S_n, \quad \forall n \in \mathcal{N} \cup \{0\}, \tag{5b}$$

$$\sum_{f\in\mathcal{F}} \sum_{u\in\mathcal{U}_n} y_n^f \lambda_{nu}^f s_f \leq C_n,\ \forall n \in \mathcal{N}, \tag{5c}$$

$$\sum_{f\in\mathcal{F}} \sum_{u\in\mathcal{U}_n} y_{kn}^f \lambda_{nu}^f s_f \leq C_{kn},\ \forall k \in \mathcal{N}, \forall n \in \mathcal{N}, \tag{5d}$$

$$y_n^f \leq x_0^f,\ \forall n \in \mathcal{N}, \forall f \in \mathcal{F}, \tag{5e}$$

$$y_{kn}^f \leq x_k^f, y_{nn}^f = 0,\ \forall n \in \mathcal{N}, \forall k \in \mathcal{N}, \forall f \in \mathcal{F}, \tag{5f}$$

$$x_n^f + y_n^f + \sum_{k\in\mathcal{N}} y_{kn}^f \leq 1,\ \forall n \in \mathcal{N}, \forall f \in \mathcal{F}, \tag{5g}$$

$$x_n^f, x_0^f, y_n^f, y_{kn}^f \in \{0,1\},\ \forall n \in \mathcal{N}, \forall k \in \mathcal{N}, \forall f \in \mathcal{F}. \tag{5h}$$

Specifically, (5b) denotes the constraints of cache sizes of all the BSs. (5c) and (5d) denote the link capacity constraints between the MBS and FBSs and among FBSs, respectively. (5e) and (5f) guarantee that a content can be delivered from the MBS or between two FBSs only when the content is cached at the MBS or the corresponding FBS. (5g) guarantees that any content will not be delivered from the MBS or another FBS if the content is locally available. The optimization objective of the problem in (5) can be rewritten as $TF_0 - \sum_{n\in\mathcal{N}} \sum_{f\in\mathcal{F}} \big[\big(x_n^f + y_n^f + \sum_{k\in\mathcal{N}} y_{kn}^f\big)\lambda_n^f\big] - \sum_{f\in\mathcal{F}} x_0^f \lambda_0^f$, where $\lambda_n^f := F \sum_{u\in\mathcal{U}_n} \lambda_{nu}^f s_f > 0$, $\forall n \in \mathcal{N} \cup \{0\}$, $\forall f \in \mathcal{F}$, and $TF_0 := \sum_{n\in\mathcal{N}\cup\{0\}} \sum_{f\in\mathcal{F}} \lambda_n^f$ are positive constants.

Remark 1: The non-caching strategy can be regarded as a special case of caching strategies by setting the cache sizes of all the BSs (i.e., $\{S_n\}_{n=0}^{N}$) to zeros. As a result, all the values of $\{x_n^f, x_0^f, y_n^f, y_{kn}^f\}$ are zeros, and the corresponding expected sum of traffic load from the Internet to the MBS in the network becomes TF_0.

The hierarchical caching optimization problem in (5) is a binary integer linear programming (BILP) problem with $F(M+1)^2$ binary variables and $FM(M+3)+3M+1$ linear constraints, which is NP-complete. However, considering that the numbers (F, M) of contents and BSs are usually very large in a real-world system, it is not feasible from the perspective of practical implementation to get the optimal solution by using exact methods, e.g., branch and bound methods [16], due to their exponential complexity. Thus, we will focus on designing low-complexity heuristic methods to achieve suboptimal solutions in the next section.

III. Designing Collaborative Hierarchical Caching Framework

In this section, based on the hierarchical caching topology in Fig. 2, we decompose the large-scale BILP problem in (5) into simpler subproblems and design the corresponding algorithms.

A. Subproblem 1

We first explore the caching at FBSs and the cooperation among FBSs to satisfy femto users' requests locally. Thus, Subproblem 1 is only to optimize $x_n^f, y_n^f, \forall k \in \mathcal{N}, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}$, and is formulated as

$$\max_{\{x_n^f, y_{kn}^f\}} \sum_{n\in\mathcal{N}} \sum_{f\in\mathcal{F}} \big(x_n^f + \sum_{k\in\mathcal{N}} y_{kn}^f\big)\lambda_n^f \tag{6a}$$

$$\textit{s.t.} \sum_{f\in\mathcal{F}} x_n^f s_f \le S_n, \quad \forall n \in \mathcal{N}, \tag{6b}$$

$$\frac{1}{F}\sum_{f\in\mathcal{F}} y_{kn}^f \lambda_n^f \le C_{kn}, \ \forall k \in \mathcal{N}, \forall n \in \mathcal{N}, \tag{6c}$$

$$y_{kn}^f \le x_k^f, y_{nn}^f = 0, \ \forall n \in \mathcal{N}, \forall k \in \mathcal{N}, \forall f \in \mathcal{F}, \tag{6d}$$

$$x_n^f + \sum_{k\in\mathcal{N}} y_{kn}^f \le 1, \ \forall n \in \mathcal{N}, \forall f \in \mathcal{F}, \tag{6e}$$

$$x_n^f, y_{kn}^f \in \{0,1\}, \ \forall n \in \mathcal{N}, \forall k \in \mathcal{N}, \forall f \in \mathcal{F}. \tag{6f}$$

To solve Subproblem 1, we develop a low-complexity distributed heuristic method to achieve suboptimal solutions. The basic idea of the proposed method is to further divide Subproblem 1 into two subproblems as follow: *1) Subproblem 1a*: Optimization of $\{x_n^f\}$, which can be formulated as $\max\limits_{\{x_n^f\}} \sum\limits_{n\in\mathcal{N}} \sum\limits_{f\in\mathcal{F}} x_n^f \lambda_n^f$, *s.t.* $(6b)$ and $(6f)$; *2) Subproblem 1b*: Optimization of $\{y_{kn}^f\}$ under achieved $\{x_n^f\}$, which can be formulated as $\max\limits_{\{y_{kn}^f\}} \sum\limits_{n\in\mathcal{N}} \sum\limits_{f\in\mathcal{F}} \sum\limits_{k\in\mathcal{N}} y_{kn}^f \lambda_n^f$, *s.t.* $(6c) - (6f)$.

Subproblem 1a can be decomposed into N single knapsack problems (SKPs) and separately solved by using the greedy algorithm in [16]. Subproblem 1b can also be solved with a greedy method. The details of the proposed greedy heuristic method for solving Subproblem 1 is shown in Algorithm 1 with the complexity $O(T\log(T))$ where $T = F \cdot N \cdot N$. In the proposed method, solving Subproblem 1a means to cache contents at each FBS in a greedy manner, while solving Subproblem 1b means to achieve the cooperation among FBSs.

Algorithm 1 Greedy Heuristic Method for Solving Subproblem 1.

```
Input: $F$, $N$, $\{\lambda_n^f\}$, $\{s_f\}$, $\{S_n\}$, $\{C_{kn}\}$.
Initialize $x_n^f = 0$, $y_{kn}^f = 0$ for $\forall n \in \mathcal{N}, \forall k \in \mathcal{N}, \forall f \in \mathcal{F}$.
–Procedure 1. [Optimize $\{x_n^f\}$]
Optimize $\{x_n^f\}$ by solving Subproblem 1a with the greedy method in [16].
–Procedure 2. [Optimize $\{y_{kn}^f\}$ under achieved $\{x_n^f\}$]
Set $\overline{C}_{kn} := C_{kn}$ for $\forall k \in \mathcal{N}, \forall n \in \mathcal{N}$, $\mathbf{A} = \boldsymbol{\pi} = \boldsymbol{\omega} := \mathbf{0}_{(FN)\times 1}$.
Reshape $\{\lambda_n^f\}$ into $\mathbf{A}$ where $A_j = \frac{1}{F}\lambda_n^f$ satisfying $f = \mathrm{mod}(j-1, F) + 1$ and $n = \frac{j-f}{F} + 1$.
Sort $\mathbf{A}$ into $\boldsymbol{\pi}$ in a descending order, and label the original indices as $\boldsymbol{\omega}$, where $A_j = \pi_i$ satisfying $j = \omega_i$.
for $i = 1$ to $FN$ do
    Calculate $j = \omega_i$, $f = \mathrm{mod}(j-1, F) + 1$ and $n = \frac{j-f}{F} + 1$.
    if $(x_n^f == 0)$ & $(\sum_{k\in\mathcal{N}} x_k^f \ge 1)$ then
        Find the set $\mathcal{S} \subseteq \{1, 2, \dots, N\}$ with all the elements w.r.t. $k$ satisfying $x_k^f = 1$, and $k^* := \arg\max_{k\in\mathcal{S}}\{\overline{C}_{kn}\}$.
        if $A_j \le \overline{C}_{k^*n}$ then
            Set $y_{k^*n}^f = 1$, $\overline{C}_{k^*n} \leftarrow \overline{C}_{k^*n} - A_j$.
        end if
    end if
end for
Output: $\{x_n^f\}$ and $\{y_{kn}^f\}$.
```

B. Subproblem 2

After achieving $x_n^f, y_{kn}^f, \forall n \in \mathcal{N}, \forall k \in \mathcal{N}, \forall f \in \mathcal{F}$ by solving Subproblem 1, Subproblem 2 is to optimize $x_0^f, y_n^f, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}$ to explore the caching at the MBS and the cooperation between the FBSs and the MBS. Specifically, from (5e) and (5g), we have $y_n^f = y_n^f \cdot y_n^f \le x_0^f z_n^f, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}$, where $z_n^f := 1 - x_n^f - \sum_{k\in\mathcal{N}} y_{kn}^f$.

Thus, to solve Subproblem 2, we first provide an approximate solution w.r.t. $\{x_0^f\}$ with the greedy method in [16] by solving the SKP as $\max\limits_{\{x_0^f\}} \sum\limits_{f\in\mathcal{F}} x_0^f e_f$, *s.t.* $\sum\limits_{f\in\mathcal{F}} x_0^f s_f \le S_0$, and $x_0^f \in \{0,1\}$, $\forall f \in \mathcal{F}$, where $e_f := \lambda_0^f + \sum_{n\in\mathcal{N}} z_n^f \lambda_n^f, \forall f \in \mathcal{F}$.

After achieving $x_0^f, \forall f \in \mathcal{F}$, we can get $y_n^f, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}$ by solving N subproblems in the form as $\max\limits_{\{y_n^f\}} \sum\limits_{f\in\mathcal{F}} y_n^f g_n^f$, *s.t.* $\frac{1}{F}\sum\limits_{f\in\mathcal{F}} y_n^f \lambda_n^f \le C_n$, and $y_n^f \in \{0,1\}$, $\forall f \in \mathcal{F}$, where $g_n^f := x_0^f z_n^f \lambda_n^f, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}$. The N subproblems are also SKPs and can be solved in parallel with the greedy method in [16]. The total complexity of solving Subproblem 2 is $O((N+1)F\log(F))$.

C. Content Request Routing

By solving the above decomposed Subproblem 1 and Subproblem 2 in order, we can get the proposed framework of collaborative hierarchical caching as well as content request routing with the practical consideration on large-scale content distribution. For any content request from femto users or macro users, the details of the proposed content request routing strategy are given in Algorithm 2.

IV. Trace-based Simulation Results

In this section, we evaluate our collaborative hierarchical caching scheme based on the traces of a real-world proxy caching system, IRCache [17], in comparison with the FemtoCaching scheme in [8]. For our simulations, we have collected the trace data for 7 days in June 2013 to represent the user requests of popular Internet contents along with their content sizes. After filtering out some illegal and incorrect data, we select $50,000$ popular contents, $4,298$ users, and

Algorithm 2 Content Request Routing Strategy.

```
**Input**: Achieved content caching results of $\{x_n^f\}$, $\{x_0^f\}$, $\{y_{kn}^f\}$ and $\{y_n^f\}$.
After receiving a request for content $o_f$, check the user type.
**if** femto user **then**
    Satisfy the request at the local FBS$_n$ if $x_n^f = 1$.
    If not yet satisfied, check the content availability of other FBSs in the macro cell, and route the request to FBS$_k$ if $y_{kn}^f = 1$.
    If not yet satisfied, route the request to the MBS, and satisfy the request if $y_n^f = 1$.
    Otherwise, download content $o_f$ directly over the Internet.
**else**
    Satisfy the request at the MBS if $x_0^f = 1$.
    Otherwise, download content $o_f$ directly over the Internet.
**end if**
```

related $516,135$ content requests. Here, user association and content requests are randomly set as modeled in Sec. II-A. We use MATLAB to implement a simulator that constructs the hierarchical caching topology, in which the macro BS is connected to M FBSs and all the FBSs are fully connected. Based on [18], we set the path loss exponent of γ as 3.7, the BSs' transmit power of (ρ_1, ρ_2) as $(40, 30)$ dBm, all the SINR threshold of (Γ_1, Γ_2) as 0 dB, the average available link capacity of (C_{kn}, C_n) as $(0.1, 1)$ Gbps. Besides, we set the ratio of the cache sizes of (S_n, S_0) as $1 : \phi$, where ϕ is a given constant. Moreover, the scalability of our proposed framework is not restricted by the aforementioned parameters.

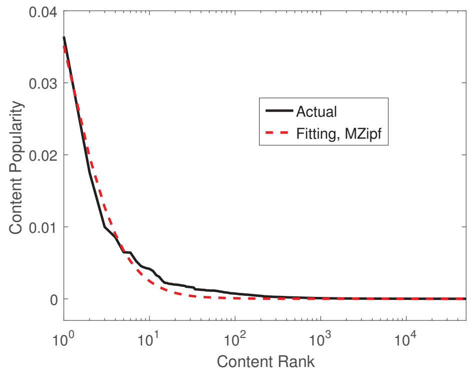

Fig. 3. Comparison of actual and fitting content popularity.

Fig. 3 compares the practical content popularity in the IRCache trace with the used theoretical content popularity model in (2). From Fig. 3, it can be observed that the practical content popularity in the trace can be well fitted by the MZipf distribution with $(\tau, \beta) = (1.56, 1.76)$, which agrees with the used model in (2). Note that the following results are based on the practical traces described above.

We then compare the performance of the FemtoCaching scheme and the proposed scheme in terms of the percentage of traffic offload and the number of supported requests versus different cache sizes of each FBS (percentage to the total content size) as shown in Fig. 4 and Fig. 5, respectively. Here, the number of FBSs is set as $M = 10$. From Fig. 4 and Fig. 5, we can observe that with the increase of the cache size of each FBS in the two schemes, both the percentage of traffic offload and the numbers of supported requests first increase significantly and then gradually go up and even stay constant. Most importantly, the proposed scheme with any ratio value of ϕ always outperforms the FemtoCaching scheme on the considered performance metrics since the hierarchical caching architecture is employed in the proposed scheme while FemtoCaching scheme uses single-tier caching architecture. Besides, as ϕ increases, the proposed scheme can achieve better performance since more storage capacity is introduced in the MBS. Specifically, from Fig. 4, the proposed scheme with four values of ϕ can respectively offload the traffic with the improvements of up to 35.3%, 49.0%, 58.4%, and 67.3% compared with the FemtoCaching scheme. From Fig. 5, the proposed scheme with four values of ϕ can respectively support up to 35.5%, 48.7%, 57.8%, and 66.8% more content requests than the FemtoCaching scheme.

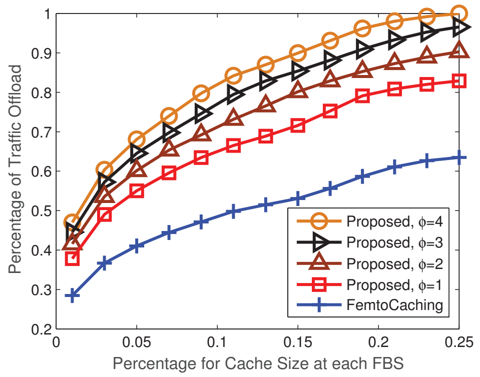

Fig. 4. Percentage of traffic offload versus different cache sizes of each FBS (percentage to the total content size) when $M = 10$.

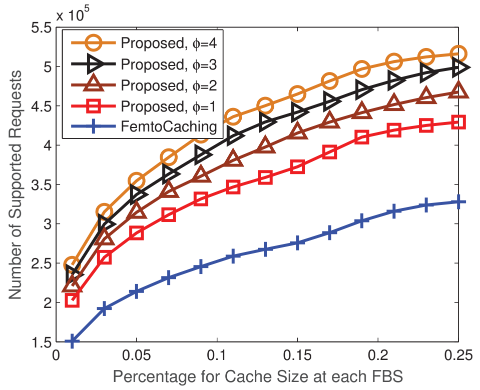

Fig. 5. Number of supported requests versus different cache sizes of each FBS (percentage to the total content size) when $M = 10$.

Fig. 6 and Fig. 7 compare the performance of the two schemes in terms of the percentage of traffic offload and the number of supported requests versus different numbers of FBSs, respectively. Here, the percentage of each FBS's cache size to the total content size is set as 5%. From Fig.

6 and Fig. 7, as the number of FBSs increases in the two schemes, both the percentage of traffic offload and the numbers of supported requests go up gradually. Most importantly, the proposed scheme with any ratio value of ϕ also achieves better performance than the FemtoCaching scheme due to the difference of used caching architectures. Besides, the proposed scheme can achieve better performance with the increase of ϕ. Specifically, from Fig. 6, compared with the FemtoCaching scheme, the proposed scheme with four cases of ϕ can offload the traffic with the improvements of up to 62.5%, 80.3%, 94.8%, and 107.4%, respectively. From Fig. 7, the proposed scheme with four cases of ϕ can respectively support up to 65.3%, 83.3%, 96.7%, and 109.2% more content requests than the FemtoCaching scheme.

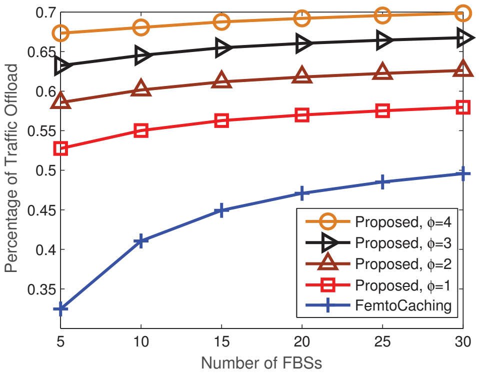

Fig. 6. Percentage of traffic offload versus different numbers of FBSs when the percentage of each FBS's cache size to the total content size is 5%.

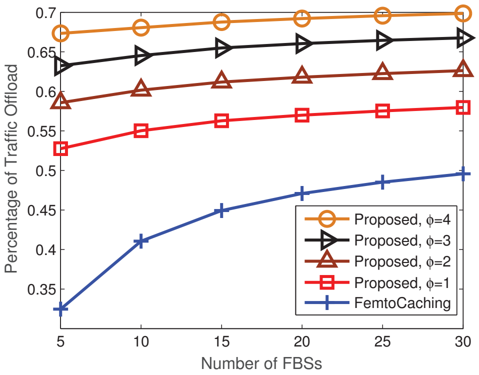

Fig. 7. Number of supported requests versus different numbers of FBSs when the percentage of each FBS's cache size to the total content size is 5%.

V. Conclusion

In this paper, we have proposed a collaborative hierarchical caching scheme in HetNets to minimize the expected sum of traffic load under the constraints of the cache sizes of the BSs and the link capacity. Based on the hierarchical caching topology, we have decomposed the formulated large-scale optimization problem into a series of simpler subproblems, and then proposed the corresponding low-complexity distributed heuristic solutions as well as a content request routing scheme. Trace-based simulation results have shown that the proposed scheme can effectively offload the traffic as well as support users' content requests.

Acknowledgment

This work is supported in part by a China Scholarship Council Four Year Doctoral Fellowship, grants from the Canadian Natural Sciences and Engineering Research Council, and the National Natural Science Foundation of China through Grant No. 61271182.

References

[1] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131-139, Feb. 2014.

[2] X. Wang, X. Li, V. C. M. Leung, and P. Nasiopoulos, "A framework of cooperative cell caching for the future mobile networks," *in Proc. HICSS*, pp. 5404-5413, Jan. 2015.

[3] X. Li, X. Wang, C. Zhu, W. Cai, and V. C. M. Leung, "Caching-as-a-Service: virtual caching framework in the cloud-based mobile networks," *in Proc. IEEE INFOCOM, Computer Communications Workshops*, pp. 372-377, May 2015.

[4] X. Li, X. Wang, S. Xiao, and V. C. M. Leung, "Delay performance analysis of cooperative cell caching in future mobile networks," *in Proc. IEEE ICC*, pp. 5652-5657, Jun. 2015.

[5] X. Li, X. Wang, and V. C. M. Leung, "Weighted network traffic offloading in cache-enabled heterogeneous networks," *in Proc. IEEE ICC*, pp. 1-6, May 2016.

[6] M. Cha, H. Kwak, P. Rodriguez, Y. Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," *In Usenix/ACM SIGCOMM IMC*, Oct. 2007.

[7] 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.

[8] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: wireless video content delivery through distributed caching helpers," *in Proc. IEEE INFOCOM*, pp. 1107-1115, Mar. 2012.

[9] Z. Zhao, M. Peng, Z. Ding, W. Wang, and H. V. Poor, "Cluster content caching: an energy-efficient approach to improve Quality of Service in cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1207-1221, May 2016.

[10] E. Baştuğ, M. Kountouris, M. Bennis, and M. Debbah, "On the delay of geographical caching methods in two-tiered heterogeneous hetworks," 2016. https://arxiv.org/abs/1605.01110.

[11] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118-6131, Sept. 2016.

[12] A. Liu and V.K.N. Liu, "Exploiting base station caching in MIMO cellular networks: opportunistic cooperation for video streaming," *IEEE Trans. Signal Processing*, vol. 63, no. 1, pp. 57-69, Jan. 2015.

[13] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Trans. Mobile Computing*, vol. PP, no. 99, Aug. 2016.

[14] H.S. Dhillon, R.K. Ganti, F. Baccelli, and J.G. Andrews, "Modeling and analysis of K-tier downlink heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 550-560, Apr. 2012.

[15] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1447-1460, Dec. 2008.

[16] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John & Sons Inc. NY, USA, 1990.

[17] National Laboratory for Applied Network Research, Weekly Squid HTTP Access Logs, http://www.ircache.net/.

[18] 3GPP, "Further advancements for E-UTRA physical layer aspects," *Tech. Rep. v.9.0.0*, Mar. 2010. http://www.3gpp.org/.