Distributed Caching for Data Dissemination in the Downlink of Heterogeneous Networks

Jun Li, *Member, IEEE*, Youjia Chen, Zihuai Lin, *Senior Member, IEEE*, Wen Chen, *Senior Member, IEEE*, Branka Vucetic, *Fellow, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—Heterogeneous cellular networks (HCNs) with embedded small cells are considered, where multiple mobile users wish to download network content of different popularity. By caching data into the small-cell base stations, we will design distributed caching optimization algorithms via belief propagation (BP) for minimizing the downloading latency. First, we derive the delay-minimization objective function and formulate an optimization problem. Then, we develop a framework for modeling the underlying HCN topology with the aid of a factor graph. Furthermore, a distributed BP algorithm is proposed based on the network's factor graph. Next, we prove that a fixed point of convergence exists for our distributed BP algorithm. In order to reduce the complexity of the BP, we propose a heuristic BP algorithm. Furthermore, we evaluate the average downloading performance of our HCN for different numbers and locations of the base stations and mobile users, with the aid of stochastic geometry theory. By modeling the nodes distributions using a Poisson point process, we develop the expressions of the average factor graph degree distribution, as well as an upper bound of the outage probability for random caching schemes. We also improve the performance of random caching. Our simulations show that 1) the proposed distributed BP algorithm has a near-optimal delay performance, approaching that of the high-complexity exhaustive search method; 2) the modified BP offers a good delay performance at low communication complexity; 3) both the average degree distribution and the outage upper bound analysis relying on stochastic geometry match well with our Monte-Carlo simulations; and 4) the optimization based on the upper bound provides both a better outage and a better delay performance than the benchmarks.**

Manuscript received December 24, 2014; revised May 9, 2015 and July 4, 2015; accepted July 7, 2015. This work was supported in part by Australian Research Council Programs under Grant DP120100405, by the National 973 Project under Grant 2012CB316106, by the NSF China under Grants 61328101, 61271230, and 61472190, by the STCSM Science and Technology Innovation Program under Grant 13510711200, and by the SEU National Key Laboratory on Mobile Communications under Grants 2013D11 and 2013D02. The associate editor coordinating the review of this paper and approving it for publication was Z. Dawy.

J. Li is with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: jleesr80@gmail.com).

Y. Chen, Z. Lin, and B. Vucetic are with the School of Electrical and Information Engineering, The University of Sydney, Sydney, N.S.W. 2006, Australia (e-mail: youjia.chen@sydney.edu.au; zihuai.lin@sydney.edu.au; branka.vucetic@sydney.edu.au).

W. Chen is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wenchen@sjtu.edu.cn).

L. Hanzo is with Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: lh@ecs.soton.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/TCOMM.2015.2455500

***Index Terms*—Wireless caching, heterogeneous cellular networks, belief propagation, stochastic geometry.**

I. Introduction

Wireless data traffic is expected to increase by a factor of 40 over the next five years, from the current level of 93 Petabytes to 3600 Petabytes per month [1], driven by a rapid increase in the number of mobile users (MU) and aggravated by their bandwidth-hungry mobile applications. A promising approach to enhancing the network capacity is to embed small cells relying on low-power base stations (BS) into the existing macro-cell based networks. These networks, which are referred to as heterogeneous cellular networks (HCN) [2]–[7], typically contain regularly deployed macro-cells and embedded femtocells as well as pico-cells [8]–[10] that are served by macro-cell BSs (MBS) and small-cell BSs (SBS), respectively. The aim of these flexibly deployed low-power SBSs is to eliminate the coverage holes and to increase the capacity in hot-spots.

There is evidence that the MUs' downloading of video on-demand files is the main reason for the growth of data traffic over cellular networks [11]. According to the prediction of Cisco on mobile data traffic, the mobile video streaming traffic will occupy 72% percentage of the overall mobile data traffic by 2019. Often, there are numerous repetitive downloading requests of popular contents, such as online blockbusters, leading to redundant data streaming. The redundancy of data transmissions can be reduced by locally storing popular data, known as caching, into the local SBSs, effectively forming a local cloud caching system (LCCS). The LCCS brings the content closer to the MUs and alleviates redundant data transmissions via redirecting the downloading requests to local SBSs. Also, the SBSs are willing to cache files into their buffers as long as they can, since caching is capable of significantly reducing the tele-traffic load on their back-haul channels, which are expensive.

In [12], the authors study the caching strategies of delay-tolerant vehicular networks, where the data subscribers and "helpers" are always moving and the links between them are opportunistic. By proposing an efficient algorithm to carefully allocate the network resources to mobile data, the decision is made as to which content should use the erasure coding, as well as conceiving the coding policy for each mobile data. In [13], optimal cache replacement policies are investigated. The cache replacement process takes place after the data caching process has been completed, and determines which particular data item should be deleted from the cache, when the available storage space is insufficient for accommodating an item to be cached.

Since the HCN structure has been widely adopted in current cellular networks and will prevail in near-future networks, we are interested in the SBS-based LCCS in the context of HCNs. In contrast to the vehicular networks discussed in [12], [14], where the mobility and the opportunistic communication contact are important issues, in the context of HCNs, the BSs are always fixed, and the MUs are assumed to be moving at a low speed. Thus, we ignore the mobility issues in the HCNs and assume that each MU is associated with a fixed BS during file-downloading. At the time of writing, there are already technical reports highlighting the advantages of caching in HCNs [15]–[17]. Based on these reports, the LCCS with SBS caching for HCNs is capable of efficiently 1) reducing the transmission latency due to short distance between the SBSs and the MUs, 2) offloading redundant data streams from MBSs, and 3) alleviating heavy burdens on the back-haul channels of the SBSs. Therefore, SBS-based caching will bring about significant breakthroughs for future HCNs.

The concept of caching is common in wireline networks and computer systems. However, research on efficient caching design for wireless cellular networks relying on small cells is still in its infancy [11], [18]. Usually, data caching consists of two phases: data placement and data transmission. During the data placement phase, data is cached into local SBSs in order to form an LCCS. In the data transmission phase, MUs request data from the LCCS. The focus of wireless caching research is mainly on the optimization of data placement for ensuring that the downloading latency is minimized. The caching optimization is a non-trivial problem. This is due to the massive scale of video contents to be stored in the limited memory of the SBSs.

The survey papers [11], [18] report on a range of attractive caching architectures conceived for future cellular networks. In [19], a caching scheme is proposed for a device-to-device (D2D) based cellular network on the MUs' caching of popular data. In this scheme, the D2D cluster size was optimized for reducing the downloading delay. In [20], [21], the authors propose a caching scheme for wireless sensor networks, where the protocol model of [22] is adopted. In [23], a femto-caching scheme is proposed for a cellular network combined with SBSs, where the data placement at the SBSs is optimized in a centralized manner for reducing the transmission delay imposed. However, [23] considers an idealized system, where neither the interference nor the impact of wireless channels is taken into account. The associations between the MUs and the SBSs are pre-determined without considering the specific channel conditions encountered. Furthermore, this centralized optimization method assumes that the MBS has perfect knowledge of all the channel state information (CSI) between the MUs and SBSs, which is impractical.

Against this background, in this paper, we consider distributed caching solutions for HCNs operating under more practical considerations. Our contributions consist of two parts.

1) In the first part, we propose distributed caching algorithms for enhancing the downloading performance via belief propagation (BP) [24]. The BP algorithm is capable of decomposing a global optimization problem into multiple sub-problems, thereby offering an efficient distributive approach of solving the global optimization problem [25]–[27]. As the BP method has been widely adopted for distributively solving resource allocation in cellular networks, we arrange file placement via BP algorithms by viewing files as a type of resource.
2) In the second part, we analyze the average caching performance based on stochastic geometry theory [28], [29]. We are interested in optimizing the average performance of a set of HCNs, where the channels exhibit Rayleigh fading and the distributions of network nodes obey a Poisson point process (PPP) [30].

Specifically, our contributions in the first part are follows.

1) We commence by deriving the delay as our optimization objective function (OF) and formulate the problem as optimizing the file placement.
2) We develop a framework for modeling the associated factor graph based on the topology of the network. A distributed BP algorithm is proposed based on the factor graph, which allows the file placement to be optimized in a distributed manner between the MUs and SBSs.
3) We prove that a fixed point exists in the proposed BP algorithm and show that the BP algorithm is capable of converging to this fixed point under certain conditions.
4) To reduce the communication complexity, we propose a heuristic BP algorithm.

Our contributions in the second part are follows.

1) By following the stochastic geometry framework, we model the MUs and SBSs in the HCN as different ties of a PPP. Furthermore, we develop the average degree distribution of the factor graph in the BP algorithm.
2) A random caching scheme is proposed, where each SBS will cache a file with a pre-determined probability. We can characterize the average downloading performance by outage probability (OP) and develop a tight upper bound of the OP expression with a closed form under the random caching scheme.
3) Based on the upper bound derived, we further improve the OP performance of random caching by optimizing the probabilities for caching different files.

In the simulations, we first investigate the average degree distribution of the factor graph, as well as the OP and the delay of the random caching schemes, in conjunction with various PPP parameters and power settings. It is shown that both the degree distribution and our upper bound analysis match well with the results of Monte-Carlo simulations. Furthermore, the optimization based on the upper bound provides both a better OP and a better delay than the benchmarks. Then we evaluate the distributed BP algorithm in our HCNs having a fixed number of BSs and MUs. It is shown that the proposed distributed BP algorithm has a near-optimal performance, approaching that of the exhaustive search method. The heuristic BP also offers a relatively good performance, despite its significantly reduced communication complexity.

The rest of this paper is organized as follows. We describe the system model in Section II and present the distributed file downloading problem relying on caching in Section III. We then propose a distributed BP algorithm in Section IV, where the proof of existence for a fixed point is also presented. In Section V, a heuristic BP algorithm is proposed for reducing the associated communication complexity. Our stochastic geometry based analysis is detailed in Section VI, where the average degree distribution of the factor graph and the OP of the random caching scheme are developed. Our simulation results are summarized in Section VII, while our conclusions are provided in Section VIII.

II. System Model

Let us consider an HCN consisting of a single MBS and K SBSs illuminating both femto-cells and pico-cells, while supporting J MUs randomly located in the network. Let us denote by $\mathcal{B}_0$ the MBS and by $\boldsymbol{\mathcal{B}} = \{\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_K\}$ the set of the SBSs, where $\mathcal{B}_k$, $k \in \boldsymbol{\mathcal{K}} = \{1, 2, \cdots, K\}$, represents the k-th SBS. Furthermore, denote by $\boldsymbol{\mathcal{U}} = \{\mathcal{U}_1, \mathcal{U}_2, \cdots \mathcal{U}_J\}$ the set of the MUs, where $\mathcal{U}_j$, $j \in \boldsymbol{\mathcal{J}} = \{1, 2, \cdots, J\}$, represents the j-th MU. The MBS $\mathcal{B}_0$ caches files into the memories of the SBSs during off-peak time via back-haul channels. Once the caching process is completed, the MBSs and SBSs are ready to act upon the downloading requests of the MUs.

We assume that a dedicated frequency band of bandwidth W is allocated to the downlink channels spanning from the SBSs to the MUs for file-dissemination. For reasons of careful load balancing, we consider the "SBS-first" constraint, where each MU will try to download data from its adjacent SBSs, unless the required files cannot be found in these SBSs. In this case, the MU will turn to the MBS for retrieving the required files. For the sake of simplicity, we assume that the MBS will support a fixed download rate, denoted by C_0, for the MUs in the channels which are orthogonal to those spanning from the SBSs to MUs.

In order to satisfy the "SBS-first" constraint for offloading data from the MBS, some incentives may be provided for the MUs. For example, downloading from the SBSs is much cheaper than from the MBS. Here, we assume that the download rate C_0 supported by the MBS is never higher than the lowest download rate supported by the SBSs. This limit imposed on the download rate from the MBS will not only encourage the MUs to download from the SBSs first, but also effectively control the data traffic of the MBS imposed by file downloading.

Denote by P_k the transmission power of the k-th SBS, and by σ^2 the noise power at each MU. The path-loss between $\mathcal{B}_k$ and the MU $\mathcal{U}_j$ is modeled as $d_{k,j}^{-\alpha}$, where $d_{k,j}$ is the distance between $\mathcal{B}_k$ and $\mathcal{U}_j$, and α is the path-loss exponent. The random channel between $\mathcal{B}_k$ and $\mathcal{U}_j$ is Rayleigh fading, whose coefficient $h_{k,j}$ has the average power of one. We assume that all the downlink channels spanning from the SBSs to the MUs are independent and identically distributed (i.i.d.).

Suppose that each file is split into multiple chunks and each chunk can be downloaded by an MU in a short time slot. Due to the short downloading time of a chunk, we assume furthermore that the probability of having two MUs streaming a chunk at the same time (or within a relative delay of a few seconds) from the same SBS is basically zero [20]. Hence, neither direct multicasting by exploiting the broadcast nature of the wireless medium nor network coding is considered. Furthermore, we focus our attention on the saturated scenario, where the SBSs keep transmitting data to the MUs [31]. Hence, each MU is subject to the interference imposed by all the other SBSs in $\boldsymbol{\mathcal{B}}$, when downloading files from its associated SBS. Given a channel realization $\mathbf{h}_j = [h_{1,j}, \cdots, h_{K,j}]$, the channel capacity between $\mathcal{B}_k$ and $\mathcal{U}_j$ can be calculated based on the signal-to-interference-plus-noise ratio (SINR) as

$$C_{k,j} = W \log \left(1 + \frac{h_{k,j}^2 d_{k,j}^{-\alpha} P_k}{\sum\limits_{q \in \boldsymbol{\mathcal{K}} \setminus \{k\}} h_{q,j}^2 d_{q,j}^{-\alpha} P_q + \sigma^2} \right). \tag{1}$$

Due to the 'SBS-first' constraint, we have $C_0 \leq C_{k,j}$, $\forall k \in \boldsymbol{\mathcal{K}}, j \in \boldsymbol{\mathcal{J}}$.

Denote by $\boldsymbol{\mathcal{F}}$ the library or set of files, which consists of Q popular files to be requested frequently by the MUs. The popularity distribution among the set $\boldsymbol{\mathcal{F}}$ is represented by $\boldsymbol{\mathcal{P}} = \{p_1, p_2, \cdots, p_Q\}$, where the MUs make independent requests of the f-th file, $f = 1, \cdots, Q$, with the probability of p_f. Without any loss of generality, all these files have the same size of M bits. We assume that $\mathcal{B}_0$ has a sufficiently large memory and hence accommodates the entire library of files, while the storage of each SBS is limited to G files, where we have $G < Q$.

Without a loss of generality, we assume that Q/G is an integer. The Q files in $\boldsymbol{\mathcal{F}}$ are divided into $N = Q/G$ file groups (FG), with each FG containing G files. The f-th file, $\forall f \in \{(n-1)G + 1, \cdots, nG\}$, is included in the n-th FG, $n \in \boldsymbol{\mathcal{N}} = \{1, \cdots, N\}$. We denote by $\mathcal{F}_n$ the n-th FG, and by $P_{\mathcal{F}_n}$ the probability that the MUs request a file in $\mathcal{F}_n$. Based on $\boldsymbol{\mathcal{P}}$, we have

$$P_{\mathcal{F}_n} = \sum_{f=(n-1)G+1}^{nG} p_f. \tag{2}$$

File caching is then carried out on the basis of FG, i.e., each SBS caches one of the N FGs.

III. Distributed File Downloading Relying on Caching

The caching-based distributed file downloading protocol consists of two stages. The first stage, or file placement stage, includes file content broadcasting and caching. In this stage, $\mathcal{B}_0$ broadcasts the FGs to the SBSs via the back-haul during off-peak periods. At the same time, the SBSs listen to the broadcasting from $\mathcal{B}_0$, and cache the FGs needed. The second stage, or file downloading stage, includes MU-SBS associations and file content transmissions. In this stage, each MU makes decisions as to which SBSs it should be associated with, and then starts to download files from the associated SBSs. When the requested files are not found in the adjacent SBSs, the MUs will turn to the MBS for these files.

A. File Placement Matrix

For assigning the N FGs to the K SBSs, we set up a file placement matrix $\boldsymbol{\Lambda}$ of size $K \times N$. The entry $\lambda_{k,n} \in \{0, 1\}$ in $\boldsymbol{\Lambda}$ indicates whether $\mathcal{F}_n$ is cached by $\mathcal{B}_k$ or not. We have $\lambda_{k,n} = 1$ if $\mathcal{F}_n$ is cached by $\mathcal{B}_k$, while $\lambda_{k,n} = 0$ otherwise. The

k-th row of $\mathbf{\Lambda}$ indicates which FG is cached by $\mathcal{B}_k$, and the n-th column indicates which BS caches $\mathcal{F}_n$. The number of the SBSs which cache $\mathcal{F}_n$ can be calculated as $\sum_{k\in\boldsymbol{\mathcal{K}}}\lambda_{k,n}$. Since each SBS caches one FG, we have $\sum_{n\in\boldsymbol{\mathcal{N}}}\lambda_{k,n}=1$.

B. MU-SBS Association

Denote by $\boldsymbol{\mathcal{H}}(j)$ the subscript set of the specific SBSs, which are capable of providing a sufficiently high SINR for the MU $\mathcal{U}_j$. The SBSs in $\boldsymbol{\mathcal{H}}(j)$ are the candidates for $\mathcal{U}_j$ to be potentially associated with. By setting an SINR threshold δ, $\mathcal{B}_k$ will be included in $\boldsymbol{\mathcal{H}}(j)$ if and only if

$$\frac{h_{k,j}^2 d_{k,j}^{-\alpha} P_k}{\sum\limits_{q\in\boldsymbol{\mathcal{K}}\backslash\{k\}} h_{q,j}^2 d_{q,j}^{-\alpha} P_q+\sigma^2}\geq\delta. \tag{3}$$

When requesting a file in $\mathcal{F}_n$, $\mathcal{U}_j$ first communicates with one of the SBSs in $\boldsymbol{\mathcal{H}}(j)$ which caches $\mathcal{F}_n$. It is possible that more than one SBS in $\boldsymbol{\mathcal{H}}(j)$ caches $\mathcal{F}_n$. In this case, $\mathcal{U}_j$ will associates with the optimal SBS, which imposes the minimum downloading delay.

It is clear that the downloading delay is inversely proportional to the downlink transmission rate. According to the file request assumption stipulated in the previous section, there is only a single MU connected to an SBS at each time. Thus, the maximum transmission rate from $\mathcal{B}_h$ to $\mathcal{U}_j$, $\forall\, h\in\boldsymbol{\mathcal{H}}(j)$, is the channel capacity between them, i.e., $C_{h,j}$. When $\mathcal{U}_j$ tries to download a file in $\mathcal{F}_n$, it follows the maximum-capacity association criterion. Hence, $\mathcal{U}_j$ associates with $B_{\hat{h}}$ such that

$$\hat{h}=\mathop{\arg\max}_{h\in\boldsymbol{\mathcal{H}}(j)}\left\{\lambda_{h,n}C_{h,j}\right\}. \tag{4}$$

When none of the SBSs in $\boldsymbol{\mathcal{H}}(j)$ caches $\mathcal{F}_n$, i.e., we have $\lambda_{h,n}=0$, $\forall\, h\in\boldsymbol{\mathcal{H}}(j)$, $\mathcal{U}_j$ will associate with the MBS for the requested file.

C. Optimization Problem Formulation

We now optimize the matrix $\mathbf{\Lambda}$ for minimizing the average delay of downloading a file. Only when the optimal $\mathbf{\Lambda}$ has been determined will the file-placement stage commence, where the files are placed according this optimal matrix. Once the MU-SBS associations have been determined, we can optimize the matrix $\mathbf{\Lambda}$ for minimizing the average delay of downloading a file. First, given the channel coefficients and the specific location of $\mathcal{U}_j$, the delay of downloading a file in $\mathcal{F}_n$ by $\mathcal{U}_j$ can be calculated as

$$D_{j,n}=\begin{cases}\frac{M}{\max_{h\in\boldsymbol{\mathcal{H}}(j)}\{\lambda_{h,n}C_{h,j}\}}, & \exists\lambda_{h,n}\neq 0,\quad \forall\, h\in\boldsymbol{\mathcal{H}}(j)\\ \frac{M}{C_0}, & \text{otherwise.}\end{cases} \tag{5}$$

Based on the request probability of each FG, the delay for $\mathcal{U}_j$ to download a file from $\boldsymbol{\mathcal{F}}$ can be written as $D_j=\sum_{n\in\boldsymbol{\mathcal{N}}}P_{\mathcal{F}_n}D_{j,n}$. Thus, the average delay for each MU can be calculated as

$$D=\frac{1}{J}\sum_{j\in\boldsymbol{\mathcal{J}}}D_j. \tag{6}$$

By setting D as the OF, let us hence formulate the delay optimization problem as follows:

$$\begin{aligned}&\text{minimize} \quad D\\ &\text{s.t.} \qquad \sum_{n\in\boldsymbol{\mathcal{N}}}\lambda_{k,n}=1,\quad \forall\, k\in\boldsymbol{\mathcal{K}},\\ &\qquad\qquad \mathbf{\Lambda}\in\{0,1\}^{K\times N}.\end{aligned} \tag{7}$$

The optimization problem in (7) is an integer programming problem, which is NP-complete. In [14], [23], similar optimization problems have been solved by sub-optimal solutions, such as the classic greedy algorithm (GA). However, the existing solutions are typically based on centralized optimization. As we can see from (6), a centralized minimization of D at $\mathcal{B}_0$ requires the global CSI between $\boldsymbol{\mathcal{B}}$ and $\boldsymbol{\mathcal{U}}$, which is impractical. Hence, we will dispense with this assumption and optimize $\mathbf{\Lambda}$ in a distributed manner at a low complexity.

IV. Distributed Belief Propagation Algorithm

In this section, we propose a distributed algorithm based on BP for solving the optimization problem of (7) as follows: 1) We first develop a factor graph for describing the message passing in the BP algorithm. 2) Then we map the resultant factor graph to the network for the sake of facilitating the distributed BP optimization. 3) This solved by solving our optimization problem by proposing a distributed BP algorithm. 4) Finally, the proof of existence for a fixed point of convergence in the BP algorithm is presented.

A. Factor Graph Model

In our BP algorithm, the factor graph has to be first established based on the underlying network as a standard bipartite graphical representation of a mathematical relationship between the local delay functions and file allocation variables. Then the BP algorithm is implemented by iteratively passing messages between the local functions and their related variables. Our optimization problem is thus solved by the proposed BP algorithm based on the factor graph.

Based on the topology of the HCN, we develop a factor graph model $\boldsymbol{\mathcal{G}}=(\boldsymbol{\mathcal{V}},\boldsymbol{\mathcal{E}})$, where $\boldsymbol{\mathcal{V}}$ is the vertex set, and $\boldsymbol{\mathcal{E}}$ is the edge set. The vertex set $\boldsymbol{\mathcal{V}}$ consists of factor nodes and variable nodes. Each factor node is related to an MU and each variable node is related to an SBS. To simplify the notations, we denote by $j\in\boldsymbol{\mathcal{J}}$ the j-th factor node and denote by $k\in\boldsymbol{\mathcal{K}}$ the k-th variable node. Hence, the vertex set $\boldsymbol{\mathcal{V}}$ is composed of $\boldsymbol{\mathcal{J}}$ and $\boldsymbol{\mathcal{K}}$, i.e., $\boldsymbol{\mathcal{V}}=\{\boldsymbol{\mathcal{J}},\boldsymbol{\mathcal{K}}\}$.

As mentioned in the previous section, $\mathcal{B}_k$ will be a candidate for $\mathcal{U}_j$ to potentially associate with, but only if the received SINR at $\mathcal{U}_j$ from $\mathcal{B}_k$ is no less than the threshold δ. Correspondingly, in our factor graph, an edge in the edge set $\boldsymbol{\mathcal{E}}$ connecting $\mathcal{U}_j$ and $\mathcal{B}_k$, denoted by (j,k), exists if the received SINR at $\mathcal{U}_j$ from $\mathcal{B}_k$ is no less than δ. The node k is named as a neighboring node of j, if there is an edge (j,k). Actually,

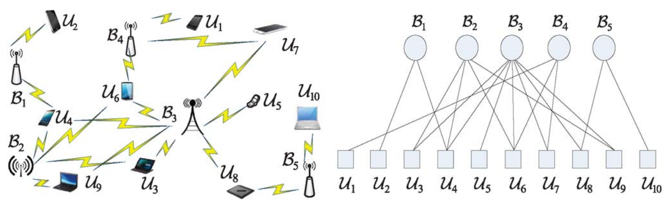


Fig. 1. Factor graph extracted from an HCN composed of 5 SBSs and 10 MUs. The edge between an SBS and an MU means that the SBS can provide a sufficiently high SINR for the MU. For instance, $\mathcal{B}_1$ can provide a sufficiently high SINR for $\mathcal{U}_2$ as well as $\mathcal{U}_4$. At the same time, $\mathcal{U}_3$ can receive a sufficiently high SINR from both $\mathcal{B}_2$ and $\mathcal{B}_3$.

$\mathcal{H}(j)$ defined previously represents the set of the neighboring nodes of the factor node j. Furthermore, denote by $\mathcal{H}(k)$ the set of neighboring node for the variable node k. Fig. 1 illustrates a factor graph extracted from an HCN with 5 SBSs and 10 MUs. Take $\mathcal{B}_1$ in the factor graph for example. The edges exist between $\mathcal{B}_1$ and $\mathcal{U}_2$ as well as $\mathcal{U}_4$, which means that $\mathcal{B}_1$ can provide a sufficient large SINR for both $\mathcal{U}_2$ and $\mathcal{U}_4$.

The distributed BP algorithm is based on the factor graph $\mathcal{G}$. The factor nodes in $\mathcal{J}$ represent the local utility functions generated from the decomposition results of the global utility function, which will be discussed later in this subsection. The variable nodes in $\mathcal{K}$ represent the variables to be optimized, i.e., the entries of $\mathbf{\Lambda}$. The factor nodes and variable nodes are connected by edges in $\mathcal{E}$, indicating the message flows in the BP algorithm. That is, messages are only passing between a node and its neighbors. We now illustrate the optimization problem on the factor graph.

1) Factor Nodes: According to Eq. (7), the OF can be decomposed into J local contributions as $D_1, \cdots, D_J$. These local contributions are calculated based on Eq. (5). Since the BP algorithm solves maximization problems, we define a series of utility functions as $F \triangleq -D$ and $F_j \triangleq -D_j$. Then our optimization problem can be rewritten as

$$\max_{\mathbf{\Lambda}} F(\mathbf{\Lambda}), \quad F = \frac{1}{J}\sum_{j\in\mathcal{J}} F_j. \tag{8}$$

We use the j-th factor node to represent the j-th local utility function F_j, which is related to $\mathcal{U}_j$. Hence, the maximization of F can be achieved by maximizing F_j at $\mathcal{U}_j, \forall j \in \mathcal{J}$.

2) Variable Nodes: Each variable node is related to an SBS. Here, we use the k-th variable node to represent the k-th row of $\mathbf{\Lambda}$, denoted by $\boldsymbol{\lambda}_k$, which is related to $\mathcal{B}_k$. The location of '1' in $\boldsymbol{\lambda}_k$ indicates which specific FG is stored by $\mathcal{B}_k$. Note that the first constraint in (7) means that each SBS only stores a single FG. Given this constraint, $\boldsymbol{\lambda}_k$ has N possible values according to N different locations of '1'. We denote by $\boldsymbol{\lambda}_k^{[1]}, \cdots, \boldsymbol{\lambda}_k^{[N]}$ the N values of $\boldsymbol{\lambda}_k$. When we have $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]}$, this implies that the FG $\mathcal{F}_n$ is stored by $\mathcal{B}_k$. Take $N = 2$ for example, where $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[1]} = [1\ 0]$ indicates that the FG $\mathcal{F}_1$ is stored in the SBS $\mathcal{B}_k$, while $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[2]} = [0\ 1]$ indicates that $\mathcal{F}_2$ is stored in $\mathcal{B}_k$. The variables $\boldsymbol{\lambda}_k$, $k = 1, \cdots, K$, are the parameters to be optimized for maximizing F in (8). For simplicity, we use the matrix $\mathbf{\Lambda}$ to represent the set of the variables $\boldsymbol{\lambda}_k$ in the factor graph.

B. Distributed Belief Propagation

In standard BP, the variables are optimized by estimating their marginal probability distributions [32]. Note that the utility function F is a function of the file placement matrix $\mathbf{\Lambda}$. We define the probability mass function (PMF) $p(\mathbf{\Lambda})$ of $\mathbf{\Lambda}$ based on the utility function $F(\mathbf{\Lambda})$ as

$$p(\mathbf{\Lambda}) \triangleq \frac{1}{Z}\exp\left(\mu F(\mathbf{\Lambda})\right), \tag{9}$$

where μ is a positive number and Z is the normalization factor. According to [32], the result of large deviations shows that when $\mu \to \infty$, $p(\mathbf{\Lambda})$ concentrates around the maxima of $F(\mathbf{\Lambda})$, i.e., $\lim_{\mu\to\infty}\mathbb{E}(\mathbf{\Lambda}) = \arg\max_{\mathbf{\Lambda}} F(\mathbf{\Lambda})$, where $\mathbb{E}(\mathbf{\Lambda})$ is the expectation of $\mathbf{\Lambda}$. Once we obtain $\mathbb{E}(\mathbf{\Lambda})$, we can have a good estimate of the specific $\mathbf{\Lambda}$ which maximizes $F(\mathbf{\Lambda})$.

In our distributed BP, the maximization of F can be decomposed into J maximization operations on F_j at $\mathcal{U}_j, j = 1, \cdots, J$. Correspondingly, the estimation of $\mathbf{\Lambda}$ is decomposed into J estimations of its subsets $\mathbf{\Lambda}_j$ at $\mathcal{U}_j$, where $\mathbf{\Lambda}_j = \{\boldsymbol{\lambda}_h, \forall h \in \mathcal{H}(j)\}$. The PMF of $\mathbf{\Lambda}_j$ is written as $p_j(\mathbf{\Lambda}_j) = \frac{1}{Z_j}\exp(\mu F_j(\mathbf{\Lambda}_j))$, where Z_j is the normalization factor. Since all the variables are independent, the estimation of $\mathbf{\Lambda}_j$ at $\mathcal{U}_j$ can be further decomposed into the estimation of each individual $\boldsymbol{\lambda}_h$ via calculating its PMF $p_j(\boldsymbol{\lambda}_h)$, which is the marginal PMF of $p_j(\mathbf{\Lambda}_j)$ with respect to the variable $\boldsymbol{\lambda}_h$. Hence we have $p_j(\boldsymbol{\lambda}_h) = \mathbb{E}_{\sim\boldsymbol{\lambda}_h}(p_j(\mathbf{\Lambda}_j))$, where $\mathbb{E}_{\sim\boldsymbol{\lambda}_h}(\cdot)$ represents the expectation over the elements in $\mathbf{\Lambda}_j$, except for $\boldsymbol{\lambda}_h$. The PMF $p_j(\boldsymbol{\lambda}_h)$ is viewed as the message, which is iteratively updated between $\mathcal{U}_j$ and $\mathcal{B}_h, \forall h \in \mathcal{H}(j)$. The PMF $p_j(\boldsymbol{\lambda}_h)$ consists of N probabilities estimated by $\mathcal{U}_j$, i.e., $\Pr(\boldsymbol{\lambda}_h = \boldsymbol{\lambda}_h^{[1]}), \cdots, \Pr(\boldsymbol{\lambda}_h = \boldsymbol{\lambda}_h^{[N]})$, where $\Pr(\boldsymbol{\lambda}_h = \boldsymbol{\lambda}_h^{[n]})$ represents the probability that $\mathcal{F}_n$ is stored by $\mathcal{B}_h$.

Without a loss of generality, we assume that the edge (j, k) does exist in the factor graph. We represent the iteration index by t and denote by $p^{(t)}_{k\to j}(\boldsymbol{\lambda}_k)$ and $p^{(t)}_{j\to k}(\boldsymbol{\lambda}_k)$ the belief messages emanated from $\mathcal{B}_k$ to $\mathcal{U}_j$ and from $\mathcal{U}_j$ to $\mathcal{B}_k$ during the t-th iteration, respectively. The steps describing the distributed BP are as follows.

1) Initialization: At the variable nodes, set $t = 1$ and let $p^{(1)}_{k\to j}(\boldsymbol{\lambda}_k)$ to be the initial distribution of $\boldsymbol{\lambda}_k$, e.g., the *a priori* popularity distribution $\mathcal{P}$.

2) Variable Node Update: During the t-th iteration, each SBS $\mathcal{B}_k$ updates the message $p^{(t)}_{k\to j}(\boldsymbol{\lambda}_k)$ to be sent to $\mathcal{U}_j$ based on the messages gleaned from $\mathcal{B}_k$'s neighboring MUs other than $\mathcal{U}_j$ in the previous iteration. This includes the calculations of N probabilities. Given $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]}, \forall n \in \mathcal{N}$, we have

$$p^{(t)}_{k\to j}\left(\boldsymbol{\lambda}_k^{[n]}\right) = \frac{1}{Z_k}\prod_{\hbar\in\mathcal{H}(k)\setminus\{j\}} p^{(t-1)}_{\hbar\to k}\left(\boldsymbol{\lambda}_k^{[n]}\right), \tag{10}$$

where Z_k is the normalization factor so that we have $\sum_{n\in\mathcal{N}} p^{(t)}_{k\to j}(\boldsymbol{\lambda}_k^{[n]}) = 1$.

3) Factor Node Update: In the t-th iteration, $\mathcal{U}_j$ updates the N probabilities of the message $p^{(t)}_{j\to k}(\boldsymbol{\lambda}_k)$ to be sent to $\mathcal{B}_k$, which is based on the messages received from $\mathcal{U}_j$'s neighboring SBSs, except for $\mathcal{B}_k$. The messages updated at the factor nodes are calculated according to the marginal PMF. Given $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]}$, $\forall n \in \boldsymbol{\mathcal{N}}$, we have

$$\begin{aligned} p_{j\to k}^{(t)}\left(\boldsymbol{\lambda}_k^{[n]}\right) &= \mathbb{E}_{\sim\boldsymbol{\lambda}_k}\left(\exp\left(\mu F_j\left(\boldsymbol{\lambda}_k^{[n]}, \{\boldsymbol{\lambda}_h,\ \forall h \in \boldsymbol{\mathcal{H}}(j)\setminus\{k\}\}\right)\right)\right) \\ &= \sum_{h\in\boldsymbol{\mathcal{H}}(j)\setminus\{k\}} \sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}} \left(\prod_{q\in\boldsymbol{\mathcal{H}}(j)\setminus\{k\}} p_{q\to j}^{(t)}(\boldsymbol{\lambda}_q)\cdot\right. \\ &\qquad \left.\exp\left(\mu F_j\left(\boldsymbol{\lambda}_k^{[n]}, \{\boldsymbol{\lambda}_h,\ \forall h \in \boldsymbol{\mathcal{H}}(j)\setminus\{k\}\}\right)\right)\right). \end{aligned} \tag{11}$$

4) Final Solution: Let us assume that there are $t = T$ iterations in the distributed BP algorithm. After T iterations, the probability that $\mathcal{F}_n$ is stored by $\mathcal{B}_k$ can be obtained by

$$\Pr\left(\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]}\right) = \frac{1}{Z_k}\prod_{\hbar\in\boldsymbol{\mathcal{H}}(k)} p_{\hbar\to k}^{(T)}\left(\boldsymbol{\lambda}_k^{[n]}\right). \tag{12}$$

Based on (12), the decision as to which file should be stored by $\mathcal{B}_k$ can be made by choosing the specific file that has the maximum *a posteriori* probability $\Pr(\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]})$, $\forall n \in \boldsymbol{\mathcal{N}}$.

C. Convergence to a Fixed Point

Let us now investigate the existence of a fixed point of convergence in our distributed BP algorithm. The essence of the distributed BP algorithm is to keep updating the PMF $p_j(\boldsymbol{\lambda}_k)$ before reaching its final estimate. Based on (10) and (11), the evolution of $p_j(\boldsymbol{\lambda}_k)$ during the t-th iteration can be obtained from the PMFs in the $(t-1)$-th iteration as

$$\begin{aligned} p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k) = \frac{1}{Z_k}\prod_{\hbar\in\boldsymbol{\mathcal{H}}(k)\setminus\{j\}} \sum_{h\in\boldsymbol{\mathcal{H}}(\hbar)\setminus\{k\}} \sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}} \\ \left(\exp\left(\mu F_\hbar(\boldsymbol{\Lambda}_\hbar)\right)\cdot \prod_{q\in\boldsymbol{\mathcal{H}}(\hbar)\setminus\{k\}} p_{q\to\hbar}^{(t-1)}(\boldsymbol{\lambda}_q)\right). \end{aligned} \tag{13}$$

We view the PMF $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k)$ as a probability vector of length N. We define the probability vector set $\boldsymbol{\mathcal{M}}^{(t)} \triangleq \left\{p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k)\right\}$ for all $k \in \boldsymbol{\mathcal{K}}$ as well as $j \in \boldsymbol{\mathcal{J}}$, and define the message mapping function $\boldsymbol{\Gamma}: \mathbb{R}^{N\times KJ} \to \mathbb{R}^{N\times KJ}$ based on (13) so that $\boldsymbol{\mathcal{M}}^{(t)} = \boldsymbol{\Gamma}(\boldsymbol{\mathcal{M}}^{(t-1)})$. Then we have the following lemma.

Lemma 1: The message mapping function $\boldsymbol{\Gamma}$ is a continuous mapping.

Proof: Please refer to Appendix A.

Given Lemma 1, we have the following theorem.

Theorem 1: A fixed point of convergence exists for the proposed distributed BP algorithm.

Proof: Please refer to Appendix B.

The question of convergence to the fixed point is, unfortunately, not well understood in general [24]. Generally, if the factor graph contains no cycles, the belief propagation can be shown to converge to a fixed solution point in a finite number of iterations. The performance, including the optimality and the convergence rate, of the BP crucially depends on the choice of the objective function, as well as the scale, the sparsity and the number of cycles in the underlying factor graph. As such, the theoretical analysis of the BP algorithm's optimality and convergence rate remains an open challenge.

V. A Heuristic BP With Reduced Complexity

In the context of the BP algorithm, the message $p_j(\boldsymbol{\lambda}_k)$ exchanged between $\mathcal{U}_j$ and $\mathcal{B}_k$ in each iteration, includes N probability values, which are real numbers. Hence, the communication overhead of the message passing is relatively high. Hence, we propose a heuristic BP (HBP) algorithm for reducing the communication overhead imposed. The rationale behind the term "heuristic BP" is that we still follow the classic concept of belief propagation, but use a different format of the beliefs from the conventional one.

Assuming that the edge (j, k) exists, in the t-th iteration of the HBP, instead of forwarding the N probabilities stored in $p_{j\to k}^{(t)}(\boldsymbol{\lambda}_k)$ to $\mathcal{B}_k$, $\mathcal{U}_j$ randomly selects an FG according to these N probabilities. Then the integer index $n_{j\to k}^{(t)}$ of the FG selected will be forwarded to the SBS $\mathcal{B}_k$.

At the SBS side, the SBS $\mathcal{B}_k$ receives $|\boldsymbol{\mathcal{H}}(k)|$ integers, i.e., $n_{\hbar\to k}^{(t)}$, $\forall \hbar \in \boldsymbol{\mathcal{H}}(k)$, from its neighboring MUs, where $|\cdot|$ denotes the cardinality of a set. Based on $n_{\hbar\to k}^{(t)}$, the SBS $\mathcal{B}_k$ infers the number of those MUs, which indicate that $\mathcal{F}_n$ should be stored in the SBS $\mathcal{B}_k$, for $n = 1, \cdots, N$. Let us assume now that in the t-th iteration, there are $J_{k,n}^{(t)}$ MUs specifically indicating that $\mathcal{F}_n$ should be stored in $\mathcal{B}_k$, where we have $\sum_{n\in\boldsymbol{\mathcal{N}}} J_{k,n}^{(t)} = |\boldsymbol{\mathcal{H}}(k)|$. We can view $\frac{J_{k,n}^{(t)}}{|\boldsymbol{\mathcal{H}}(k)|}$ as the probability that the specific FG $\mathcal{F}_n$ is stored by the SBS $\mathcal{B}_k$.

In this case, the probability $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ in (10) will be recalculated as

$$p_{k\to j}^{(t)}\left(\boldsymbol{\lambda}_k^{[n]}\right) = \begin{cases} \frac{J_{k,n}^{(t-1)}-1}{|\boldsymbol{\mathcal{H}}(k)|-1}, & \text{if } n = n_{j\to k}^{(t-1)}, \\ \frac{J_{k,n}^{(t-1)}}{|\boldsymbol{\mathcal{H}}(k)|-1}, & \text{if } n \neq n_{j\to k}^{(t-1)}. \end{cases} \tag{14}$$

Note that in (14), the information $n_{j\to k}^{(t-1)}$ transmitted from the MU $\mathcal{U}_j$ to the SBS $\mathcal{B}_k$ is excluded when calculating $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$, for the sake of ensuring that only uncorrelated information is exchanged throughout the HBP.

At the MU side, it is clear that the MU $\mathcal{U}_j$ has to obtain $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ for the sake of updating the output information. However, there is no need for the SBS $\mathcal{B}_k$ to transmit the N probabilities $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ to each of its neighboring MUs. Alternatively, $\mathcal{B}_k$ broadcasts the N integers, $J_{k,1}^{(t)}, \cdots, J_{k,N}^{(t)}$ to the neighboring MUs for reducing the transmission overhead. After receiving the N integers from the SBS $\mathcal{B}_k$, the MU $\mathcal{U}_j$ calculates $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ in (14).

Based on the above discussions, the HBP algorithm can be summarized as follows.

1) Initialization: At the variable nodes, we set $t = 1$. The SBS $\mathcal{B}_k$ randomly generates $|\boldsymbol{\mathcal{H}}(k)|$ independent integers, $n_1, \cdots, n_{|\boldsymbol{\mathcal{H}}(k)|}$, according to the popularity distribution $\boldsymbol{\mathcal{P}}$. These integers are viewed as the indexes of the FGs. We then set $J_{n,k}^{(1)}$ to be the number of the integers that are equal to n.

2) Variable Node Update: In the t-th iteration, $\mathcal{B}_k$ updates and broadcasts the N integers $J_{n,k}^{(t)}$, for $n = 1, \cdots, N$, to the neighboring MUs. The resulting calculations performed on these N integers $J_{n,k}^{(t)}$ are based on the integers $n_{\hbar \to k}^{(t-1)}$, $\forall \hbar \in \boldsymbol{\mathcal{H}}(k)$, received from the neighboring MUs during the last iteration. Specifically, the n-th integer $J_{n,k}^{(t)}$ is obtained by counting the number of $n_{\hbar \to k}^{(t-1)}$ that are equal to n.

3) Factor Node Update: The MU $\mathcal{U}_j$ first calculates the probabilities $p_{h \to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$, $\forall h \in \boldsymbol{\mathcal{H}}(j)$ according to Eq. (14) based on the integers gleaned from the SBS $\mathcal{B}_h$. Then based on $p_{h \to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$, $\forall h \in \boldsymbol{\mathcal{H}}(j) \setminus \{k\}$, $\mathcal{U}_j$ calculates $p_{j \to k}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ according to Eq. (11). After obtaining the N probabilities $p_{j \to k}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$, $n = 1, \cdots, N$, $\mathcal{U}_j$ randomly chooses an FG according to these N probabilities and sends the index $n_{j \to k}^{(t)}$ of the FG to the SBS $\mathcal{B}_k$.

4) Final Solution: After T iterations, the SBS $\mathcal{B}_k$ makes the decision that the FG $\mathcal{F}_{\hat{n}}$ should be stored for ensuring that

$$\hat{n} = \arg\max_{n \in \boldsymbol{\mathcal{N}}} J_{k,n}^{(T)}. \tag{15}$$

The overhead of the HBP is significantly lower than that of the original BP introduced in the previous section. From a communication complexity perspective, in each iteration of the HBP, an SBS $\mathcal{B}_k$ broadcasts N integers, while an MU $\mathcal{U}_j$ transmits $|\boldsymbol{\mathcal{H}}(j)|$ integers. On the other hand, in the original BP, $\mathcal{B}_k$ transmits $N|\boldsymbol{\mathcal{H}}(k)|$ real numbers, while $\mathcal{U}_j$ transmits $N|\boldsymbol{\mathcal{H}}(j)|$ real numbers for each iteration. From a computational complexity perspective, in a single iteration of the HBP, the computational complexity is on the order of $O(N)$ at the SBS $\mathcal{B}_k$, and $O(|\mathcal{H}(j)|N^{|\mathcal{H}(j)|})$ at the MU $\mathcal{U}_j$. On the other hand, in the original BP, the computational complexity is $O(N|\mathcal{H}(k)|^2)$ at $\mathcal{B}_k$, and $O(|\mathcal{H}(j)|N^{|\mathcal{H}(j)|})$ at $\mathcal{U}_j$ for each iteration.

VI. Performance Analysis Based on Stochastic Geometry

In this section, we analyze both the average degree distribution of the factor graph and the average downloading performance based on stochastic geometry theory. We model the distribution of the MUs as a PPP Φ_U having the intensity of λ_U, and that of the SBSs as an independent PPP Φ_B with the intensity λ_B [31], [33]. For simplicity, we assume that all the SBSs have the same transmission power P. In the following, both the degree distribution and the downloading performance are averaged over both the channels' fading coefficients and over the PPP distributions of the nodes.

A. Average Degree Distributions of the Factor Graph

Let us now investigate the degree distribution of the factor graph averaged over PPP. Note that the degree of a factor node j is defined as the number of its neighboring variable nodes, given by the cardinality $|\boldsymbol{\mathcal{H}}(j)|$, while the degree of a variable node k is defined as the number of its neighboring factor nodes, i.e., $|\boldsymbol{\mathcal{H}}(k)|$. Then we have the following theorem.

Theorem 2: The factor nodes in the factor graph have the average degree

$$\zeta_U = 2\pi\lambda_B Z(\lambda_B, P, \alpha, \delta), \tag{16}$$

and the variable nodes have the average degree

$$\zeta_B = 2\pi\lambda_U Z(\lambda_B, P, \alpha, \delta), \tag{17}$$

where we have

$$Z(\lambda_B, P, \alpha, \delta) = \int_0^\infty \exp\left\{-\frac{2\lambda_B\pi}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) r^2 - \frac{\delta\sigma^2}{P} r^\alpha\right\} r\mathrm{d}r \tag{18}$$

and the Beta function $B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}\mathrm{d}t$.

Proof: Please refer to Appendix C.

When neglecting the noise, we have the following corollary based on *Theorem 2*.

Corollary 1: When neglecting the noise, $Z(\lambda_B, P, \alpha, \delta)$ in (18) can be rewritten as

$$Z(\lambda_B, P, \alpha, \delta) = \frac{\alpha}{4\pi\lambda_B B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)\delta^{\frac{2}{\alpha}}}. \tag{19}$$

Then we can simplify the average degree of the factor nodes in Eq. (16) to

$$\zeta_U = \frac{\alpha}{2\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)}, \tag{20}$$

and the average degree of the variable nodes in Eq. (17) to

$$\zeta_B = \frac{\lambda_U \alpha}{2\lambda_B \delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)}. \tag{21}$$

Proof: Please refer to Appendix D.

Equations (20) and (21) can be seen as approximations of (16) and (17), respectively, when the effects of the noise are neglected. These approximations are significantly accurate for the HCN, since the interference effects are dominant due to the dense deployments of the SBSs.

From (20), we can see that ζ_U is only related to δ and α, but is independent of λ_U, P and λ_B. In other words, the factor node degree has no relation with the intensities of the MUs and SBSs or with the power of the SBSs. The intuitive reason is that although increasing both the PPP intensities and the power of the SBSs can increase the total signal power, the interference also increases at the same time, which keeps the degree ζ_U of the factor nodes constant. Similarly, observe from (21) that ζ_B is independent of the power P, i.e., increasing the transmission power of the SBSs will not influence the average degree distribution of the factor graph.

Remark 1: We observe that $B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)=\pi$ when $\alpha=4$. Thus, we have closed-form expressions for ζ_U and ζ_B in (20) and (21), respectively, when $\alpha=4$.

B. Downloading Performance of Random Caching

Since the performance of BP based caching remains difficult for mathematical analysis in closed form, we propose a random caching scheme and analyze its performance based on stochastic geometry theory. The random caching is realized by randomly picking out $\Omega_{\mathcal{F}_n}\cdot K$ $(0\le\Omega_{\mathcal{F}_n}\le 1)$ SBSs from the entire set of K SBSs for caching the FG $\mathcal{F}_n$.

To evaluate the downloading performance, we first define an outage $\mathcal{Q}_n$ as the event of an MU's failing to find the FG $\mathcal{F}_n$ in its neighboring SBSs. The following theorem states an upper bound of the OP of $\mathcal{Q}_n$. As mentioned before, since the interference is the dominant factor predetermining the network performance, we ignore the noise effects in the following performance analysis to simplify our derivations.

Theorem 3: The OP for downloading a file in $\mathcal{F}_n$ can be upper-bounded by

$$\Pr(\mathcal{Q}_n)\le\frac{C(\delta,\alpha)\left(1-\Omega_{\mathcal{F}_n}\right)+A(\delta,\alpha)\Omega_{\mathcal{F}_n}}{C(\delta,\alpha)\left(1-\Omega_{\mathcal{F}_n}\right)+A(\delta,\alpha)\Omega_{\mathcal{F}_n}+\Omega_{\mathcal{F}_n}},\quad(22)$$

where we have $C(\delta,\alpha)\triangleq\frac{2}{\alpha}\delta^{\frac{2}{\alpha}}B\left(\frac{2}{\alpha},1-\frac{2}{\alpha}\right)$, $A(\delta,\alpha)\triangleq\frac{2\delta}{\alpha-2}{}_2F_1\left(1,1-\frac{2}{\alpha};2-\frac{2}{\alpha};-\delta\right)$, and ${}_2F_1$ represents the hypergeometric function.

Proof: Please refer to Appendix E.

When the path-loss exponent $\alpha=4$, we have $C(\delta,4)=\frac{\sqrt{\delta}}{2}\pi$ and $A(\delta,4)=\delta{}_2F_1\left(1,\frac{1}{2};\frac{3}{2},-\delta\right)$. It becomes clear from (22) that $\Pr(\mathcal{Q}_n)$ is only related to δ and $\Omega_{\mathcal{F}_n}$, where a higher δ leads to a higher $\Pr(\mathcal{Q}_n)$. This is because a larger δ will reduce the number of possibly eligible serving SBSs, resulting in an increase of OP. We can see that a higher $\Omega_{\mathcal{F}_n}$ leads to a lower $\Pr(\mathcal{Q}_n)$.

Let us define the averaged OP $\mathcal{Q}$ over all the files. Based on the file popularity, the OP of $\mathcal{Q}$ can be upper-bounded by

$$\begin{aligned}\Pr(\mathcal{Q})&=\sum_{n\in\mathcal{N}}P_{\mathcal{F}_n}\Pr(\mathcal{Q}_n)\\&\le\sum_{n\in\mathcal{N}}\frac{P_{\mathcal{F}_n}\left(C(\delta,\alpha)\left(1-\Omega_{\mathcal{F}_n}\right)+A(\delta,\alpha)\Omega_{\mathcal{F}_n}\right)}{C(\delta,\alpha)\left(1-\Omega_{\mathcal{F}_n}\right)+A(\delta,\alpha)\Omega_{\mathcal{F}_n}+\Omega_{\mathcal{F}_n}}.\quad(23)\end{aligned}$$

The average delay $\bar{D}$ of each MU can be obtained based on the average OP, i.e.,

$$\bar{D}=(1-\Pr(\mathcal{Q}))\,\bar{D}_s+\Pr(\mathcal{Q})\frac{M}{C_0},\quad(24)$$

where $\bar{D}_s$ is the average delay of downloading from the SBSs. The delay $\bar{D}$ can be seen as the average value of D in Eq. (6) over both the PPP and the channel fading. Note that $\bar{D}_s$ is usually challenging to calculate and does not have a closed form in the PPP analysis.

Next, we optimize $\Omega_{\mathcal{F}_n}$ for improving the downloading performance. Since we do not have a closed-form expression for $\bar{D}$, we minimize the upper bound of $\Pr(\mathcal{Q})$ in (23), i.e.,

$$\begin{aligned}&\max_{\{\Omega_{\mathcal{F}_n}\}}\sum_{n\in\mathcal{N}}\frac{P_{\mathcal{F}_n}\Omega_{\mathcal{F}_n}}{\Omega_{\mathcal{F}_n}\left(A(\delta,\alpha)-C(\delta,\alpha)+1\right)+C(\delta,\alpha)},\\&\text{s.t.}\quad\sum_{n\in\mathcal{N}}\Omega_{\mathcal{F}_n}=1,\\&\qquad\ \Omega_{\mathcal{F}_n}\ge 0.\end{aligned}\quad(25)$$

By relying on the classic Lagrangian multiplier, we arrive at the optimal solution as

$$\Omega^{\star}_{\mathcal{F}_n}=\max\left\{\frac{\sqrt{\frac{P_{\mathcal{F}_n}}{\xi}}-C(\delta,\alpha)}{A(\delta,\alpha)-C(\delta,\alpha)+1},0\right\},\quad(26)$$

where $\xi=\frac{\left(\sum_{q=1}^{n^*}\sqrt{P_{\mathcal{F}_q}}\right)^2}{(n^*C(\delta,\alpha_s)+A(\delta,\alpha_s)-C(\delta,\alpha_s)+1)^2}$, and n^* satisfies the constraint that $\Omega_{\mathcal{F}_n}\ge 0$.

VII. Simulation Results

In this section, we first focus on the HCNs associated with PPP distributed nodes, where we investigate the average degree distribution of the factor graph and the performance of the random caching scheme. Then we consider an HCN supporting a fixed number of nodes. We investigate the delay optimized by the BP algorithm and compare it to other benchmarks, including both the random caching and the optimal scheme using exhaustive search.

Note that the physical layer parameters in our simulations, such as the path-loss exponent, noise power, transmit power of the SBSs, and the intensity of the SBSs, are chosen to be practical and in line with the values set by 3GPP standards. For instance, the transmit power of an SBS is typically 2 Watt in 3GPP. The unit of power, such as noise power and transmit power, is the classic Watt. The intensities of the SBSs and MUs are expressed in terms of the numbers of the nodes per square kilometer. Unless specified otherwise, we set the path loss to $\alpha=4$, the number of files to $Q=100$, transmit power to $P=2$, and the noise power to $\sigma^2=10^{-10}$. All the simulations are executed with MATLAB. Also, we consider the performance averaged over a thousand network cases, where the locations of network nodes are uniformly distributed in each case, and randomly changed from case to case.

A. Average Degree Distributions of Factor Graph

We compare our Monte-Carlo simulations and analytical results in the HCNs at various transmission powers and node densities. Fig. 2 shows the average degree of the factor nodes with different transmission power P, SBSs' intensity λ_B, and MUs' intensity λ_U. We can see that for a given δ, the degree ζ_U remains unaffected by the specific choice of P, λ_B, and λ_U. Observe that our analytical results are consistent with the simulations. Similarly, Fig. 3 shows the average degree of

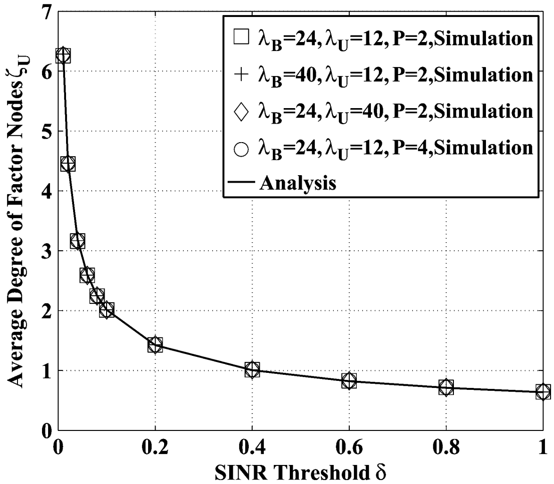


Fig. 2. Average degree of factor nodes ζ_U vs. δ for different SBS and MU intensities of λ_B and λ_U, and for transmit powers of $P = 2$ and 4.

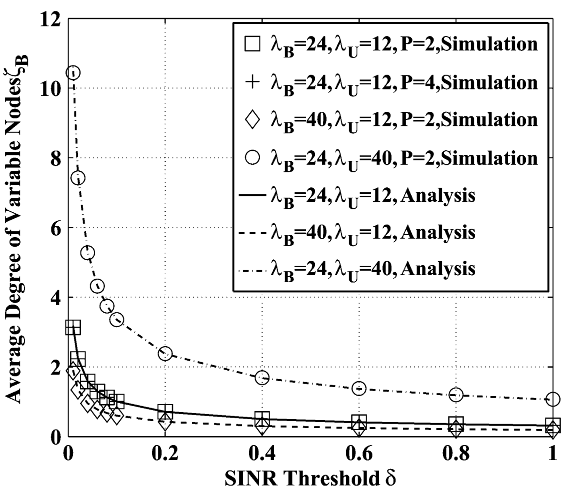


Fig. 3. Average degree of variable nodes ζ_B vs. δ for different SBS and MU intensities of λ_B and λ_U, and for transmit powers of $P = 2$ and 4.

the variable nodes of different powers and node intensities, demonstrating that the results are independent of the power P, but depend on the densities λ_B and λ_U. We can also see that the analytical results match well with the simulation results.

B. Average Downloading Performance of Random Caching

Let us now evaluate the average downloading performance of the random caching scheme supporting PPP distributed nodes. The file distribution $\mathcal{P} = \{p_1, \cdots, p_Q\}$ is modeled by the Zipf distribution [34], which can be expressed as

$$p_f = \frac{1/f^s}{\sum_{q=1}^{Q} 1/q^s}, \quad \text{for } f = 1, \cdots, Q, \tag{27}$$

where the exponent $0 < s \leq 1$ is a real number, and it characterizes the popularity of files. Explicitly, a larger s corresponds to a higher content reuse, i.e., the most popular files account for the majority of requests. Note that $P_{\mathcal{F}_n}$ can be obtained based on p_f via Eq. (2).

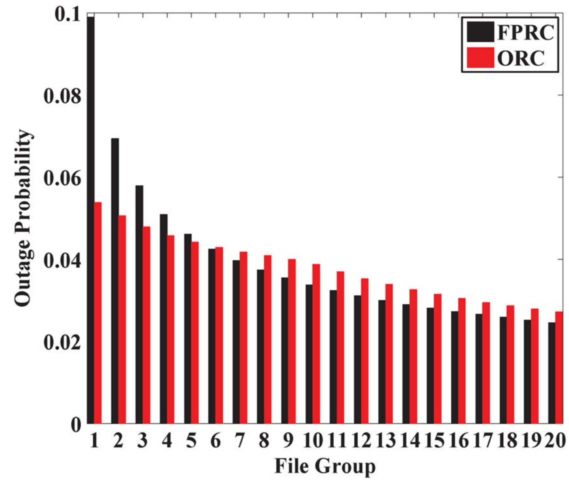


Fig. 4. Outage probabilities $\Pr(\mathcal{Q}_n) \cdot P_{\mathcal{F}_n}$ for individual FGs $\mathcal{F}_n$ under the file popularity based random caching (FPRC) and optimized random caching (ORC) schemes.

For the simulation results of this subsection, we assume that each SBS caches $G = 5$ files, hence there are $N = Q/G = 20$ FGs. We commence by considering the OP. In our optimized random caching (ORC), we set $\Omega_{\mathcal{F}_n}$ as in (26). For comparison, we also consider another random caching scheme from [19] as our the benchmark, namely, the file popularity based random caching (FPRC). In the FPRC, $\Omega_{\mathcal{F}_n}$ is chosen to be consistent with the file popularity, i.e., we have $\Omega_{\mathcal{F}_n} = P_{\mathcal{F}_n}$.

Fig. 4 shows the OPs $\Pr(\mathcal{Q}_n) \cdot P_{\mathcal{F}_n}$ for individual FGs under both the ORC and the FPRC schemes, where we have $\delta = 0.03$ and $s = 0.5$. The conditional OP $\Pr(\mathcal{Q}_n)$ (given a file in $\mathcal{F}_n$ is requested) is calculated from Eq. (22), while the request probability $P_{\mathcal{F}_n}$ of $\mathcal{F}_n$ is calculated from Eq. (2). The FGs are arranged in descending order of popularity, i.e., the first FG has the highest popularity, while the last one has the lowest popularity. We can see from the figure that compared to the FPRC, FGs having a higher popularity have a lower OP, while the ones with lower popularity have higher OPs in the ORC. For example, the OP for the most popular FG is around 0.054 in the ORC in contrast to 0.099 in the FPRC, while the probability of the least popular FG is 0.27 in the ORC in contrast to 0.25 in the FPRC. This is because the ORC is reminiscent of the classic water-filling, allocating more SBSs for caching the higher popular FGs for ensuring the minimization of the average OP.

Let us now investigate the average OP $\Pr(\mathcal{Q})$. Figs. 5 and 6 show $\Pr(\mathcal{Q})$ for different δ and s values, respectively. In Fig. 5, we fix $s = 0.5$, while in Fig. 6, we fix $\delta = 0.03$. The dashed lines with different marks are based on the simulations associated with various power and densities, while the solid lines represent the analytical upper bounds of Eq. (23). We can see that the average OP is independent of both the power P and densities λ_B and λ_U. The ORC scheme has a lower average OP than the FPRC. Furthermore, as expected, a higher SINR threshold δ leads to a higher OP, as shown in Fig. 5. At the same time, it is interesting to observe from Fig. 6 that a larger s, representing more imbalanced downloading requests on the different files, can dramatically reduce the OP. We can see that the upper bounds evaluated from Eq. (23) match the simulations quite accurately.

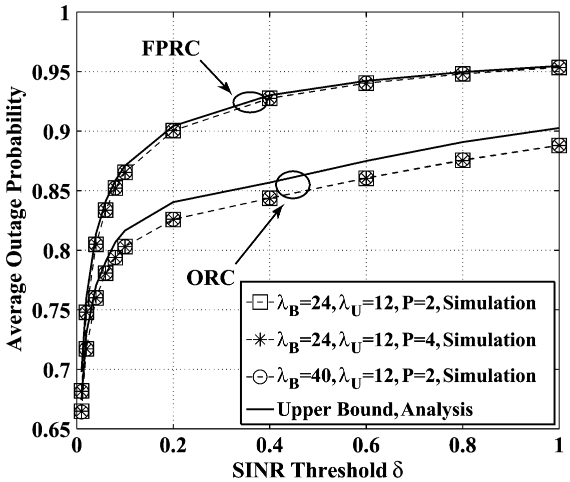


Fig. 5. Average outage probabilities $\Pr(\mathcal{Q})$ vs. δ under the FPRC and ORC schemes for different SBS and MU intensities λ_B and λ_U, and for transmit powers $P = 2$ and 4.

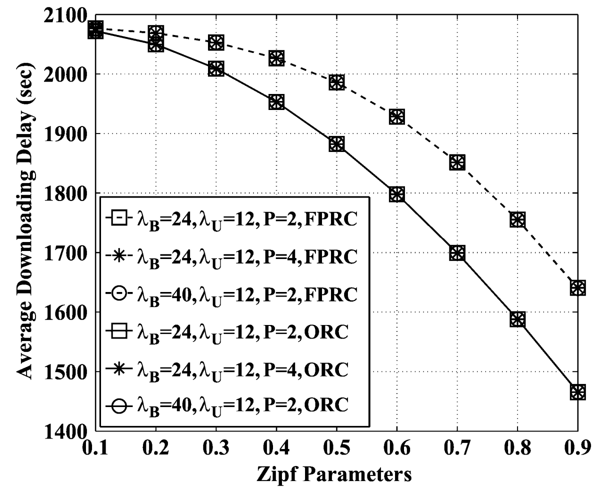


Fig. 7. Average downloading delay $\bar{D}$ vs. the Zipf parameter s under the FPRC and ORC schemes for different SBS and MU intensities λ_B and λ_U, and for transmit powers $P = 2$ and 4.

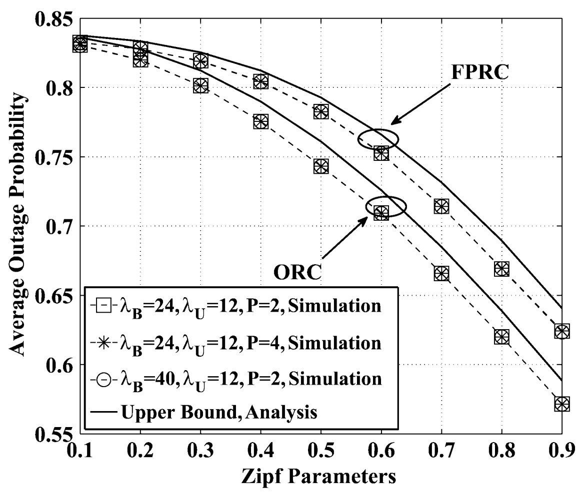


Fig. 6. Average outage probabilities $\Pr(\mathcal{Q})$ vs. the Zipf parameter s under the FPRC and ORC schemes for different SBS and MU intensities λ_B and λ_U, and for transmit powers $P = 2$ and 4.

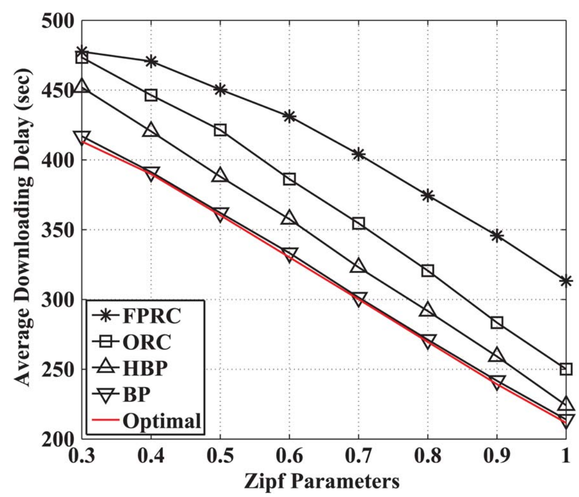


Fig. 8. Average downloading delay $\bar{D}$ vs. the Zipf parameter s under various schemes in the first scenario.

Next, we consider the average delay $\bar{D}$ in Eq. (24), where we assume an SINR threshold of $\delta = 0.03$, a bandwidth of $W = 10^7$ Hz, and a file size of $M = 10^9$ bits. Since C_0 should be always less than the maximum possible downloading rate provided by the SBSs, we assume $C_0 = W \log(1 + \delta)$. For $\delta = 0.03$, C_0 becomes 4.26×10^5 bits/sec. Fig. 7 illustrates the average downloading delay associated with different s values. We can see that the ORC scheme always outperforms the FPRC scheme, and that their performance gap becomes larger upon increasing s. Again, the observed performance does not depend on the powers and intensities of the nodes.

C. Delay Performance of Distributed BP Algorithms

Let us now study the delay performance of distributed BP-based optimizations. We consider HCNs having fixed numbers of SBSs and MUs, where the locations of these nodes are time-variant. We first consider a small network, in which the optimal solution is found with the aid of an exhaustive search. This will allow us to characterize the performance disparity between the proposed BP algorithm and the optimal search-based solution. Then we focus our attention on a larger network to show the robustness of our BP algorithms. In both scenarios, we set the SINR threshold to $\delta = 0.1$, the transmission power to $P = 2$, the bandwidth to $W = 10^7$ Hz, and the file size to $M = 10^9$ bits. Similar to the previous subsection, we assume that the rate provided by the MBS as $C_0 = W \log(1 + \delta)$. For $\delta = 0.1$, we have C_0 as 1.3×10^6 bits/sec.

In the first scenario, the nodes are arranged in a 0.6×0.6 km^2 area using 8 SBSs and 4 MUs. We assume that each SBS caches $G = 25$ files, and there are $N = Q/G = 4$ FGs. Fig. 8 shows the average delay performance under various schemes, where 'HBP' is the heuristic BP algorithm proposed in Section V, 'BP' is the original BP algorithm proposed in Section IV, and 'Optimal' is the optimal scheme relying on an exhaustive search. We can see from Fig. 8 that the original BP approaches the optimal scheme within a small delay margin. The proposed HBP performs slightly worse than the original BP, with a

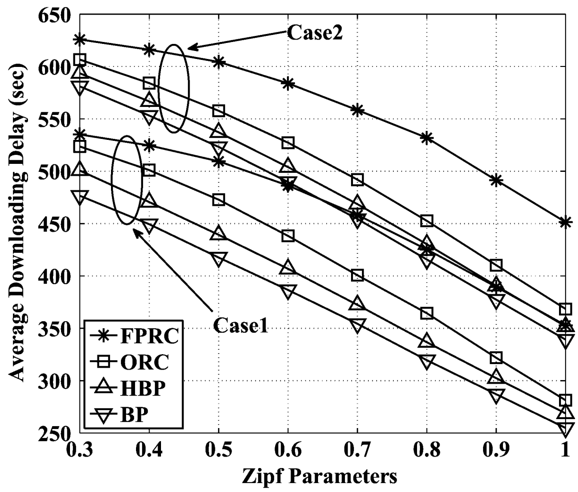


Fig. 9. Average downloading delay $\bar{D}$ vs. the Zipf parameter s under various schemes in the second scenario.

relatively modest delay degradation of around 5% or 20 seconds, while it outperforms the ORC scheme by about 10% or 40 seconds gain. The FPRC performs the worst among all the caching schemes, exhibiting a substantial delay gap between the FPRC scheme and the ORC scheme.

In the second scenario, the nodes are arranged in a 1.5×1.5 km^2 area with 50 SBSs and 25 MUs. We consider two cases, namely Case1 and Case2. In Case1, we assume that each SBS caches $G = 20$ files and there are $N = Q/G = 5$ FGs, while in Case2, we assume that each SBS caches $G = 10$ files and that we have $N = Q/G = 10$. Fig. 9 shows the average delay performance under various schemes. It is clear from Fig. 9 that in both cases the BP algorithm performs the best, while the FPRC performs the worst. The HBP exhibits a tiny delay increase of around 3% performance loss compared to the original BP, although it dramatically reduces the communication complexity during the optimization process.

Note also in Fig. 9 that the ORC suffers from a 5% performance loss compared to the HBP, but it is much less complex than the HBP and BP. The optimization in ORC is based on the statistical information available about both of channels and the locations of the nodes, while both the BP and the HBP exploit the relevant instantaneous information at a relatively high communication complexity. In this sense, the ORC constitutes an efficient caching scheme. Furthermore, we can see from Fig. 9 that there is a tradeoff between the storage and delay, i.e., a larger storage at each SBS in Case1 leads to a lower downloading delays compared to Case2.

In the above BP simulations, we set the maximum number of iterations to $T = 15$. Table I shows the average number of iterations under different s values for the two scenarios. We can see that the HBP relies on more iterations than the BP. Nevertheless, the overall communication complexity of the HBP is still lower than that of the BP, as we have discussed in Section V. Explicitly, for each iteration of the HBP, $\mathcal{B}_k$ broadcasts N integers and $\mathcal{U}_j$ transmits $|\boldsymbol{\mathcal{H}}(j)|$ integers. By contrast, in the original BP, $\mathcal{B}_k$ transmits $N|\boldsymbol{\mathcal{H}}(k)|$ real numbers and $\mathcal{U}_j$ transmits $N|\boldsymbol{\mathcal{H}}(j)|$ real numbers.

TABLE I
THE AVERAGE NUMBER OF ITERATIONS UNDER DIFFERENT s

Zipf Parameter s								
s	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Average Number of Iterations for Scenario 1								
BP	4.466	4.406	4.002	3.652	3.574	3.412	3.12	2.862
HBP	8.431	8.235	7.634	7.094	6.71	6.494	6.097	5.263
Average Number of Iterations for Scenario 2								
Case1								
BP	9.429	8.412	7.632	7.326	6.576	5.978	5.804	5.696
HBP	14.973	14.903	14.817	14.783	14.722	14.667	14.623	14.443
Case2								
BP	9.548	8.642	7.987	7.483	7.119	6.746	6.057	5.841
HBP	14.994	14.97	14.925	14.821	14.877	14.722	14.648	14.549

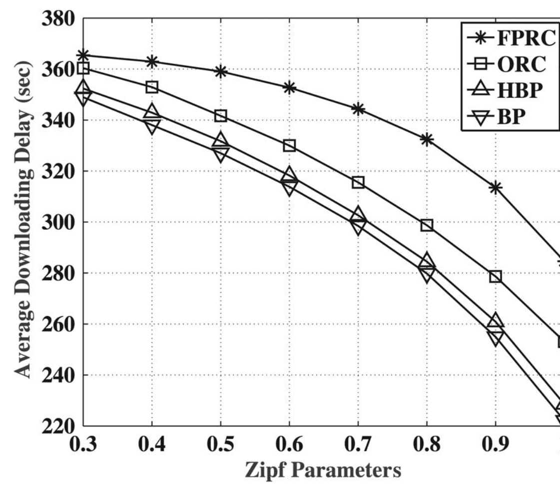


Fig. 10. Average downloading delay $\bar{D}$ vs. the Zipf parameter s under various schemes in the large scale network.

D. Delay Performance in a Large Scale Network

Finally, we consider a large-scale network associated with $Q = 1000$ files, 50 SBSs, and 100 MUs within an area of 5×5 km^2. Furthermore, we consider a lower connection probability to the SBSs by setting $\delta = 0.2$. By assuming that each SBS is capable of caching 20 files, we have overall 50 file groups. Fig. 10 shows the average delay performance. We can see from the figure that both BP algorithms perform better than the random caching schemes. Particularly, the HBP has a roughly 1% performance loss compared to the original BP, which imposes however a much reduced communication complexity. This implies that our BP algorithms are robust in large-scale networks associated with a large number of files and network nodes.

Further comparing Figs. 8, 9, and 10, it is interesting to observe that the gap between our BP and HBP algorithms becomes smaller when the network scale becomes larger. More particularly in Fig. 10, the performance of these two schemes almost overlaps. This indicate that in large scale networks, we may consider to use the HBP rather than BP to obtain a good performance at a much reduced complexity.

VIII. CONCLUSION

In this paper, we designed distributed caching optimization algorithms with the aid of BP for minimizing the downloading latency in HCNs. Specifically, a distributed BP algorithm was proposed based on the factor graph according to the network structure. We demonstrated that a fixed point of convergence exists for the distributed BP algorithm. Furthermore, we proposed a modified heuristic BP algorithm for further reducing the complexity. To have a better understanding of the average network performance under varying numbers and locations of the network nodes, we involved stochastic geometry theory in our performance analysis. Specifically, we developed the average degree distribution of the factor graph, as well as an upper bound of the OP for random caching schemes. The performance of the random caching was also optimized based on the upper bound derived. Simulations showed that the proposed distributed BP algorithm approaches the optimal performance of the exhaustive search within a small margin, while the modified BP offers a good performance at a very low complexity. Additionally, the average performance obtained by stochastic geometry analysis matches well with our Monte-Carlo simulations, and the optimization based on the upper bound derived provides a better performance than the benchmark of [19].

Appendix A
Proof of Lemma 1

To simplify the notation in the proof, we assume that $\mathcal{H}(j)=\mathcal{K}, \forall j\in\mathcal{J}$ and $\mathcal{H}(k)=\mathcal{J}, \forall\, k\in\mathcal{K}$. Consider a pair of probability vector sets $\mathcal{M}^{(t-1)}=\left\{p_{k\to j}^{(t-1)}(\boldsymbol{\lambda}_k)\right\}$ and $\widetilde{\mathcal{M}}^{(t-1)}=\left\{\tilde{p}_{k\to j}^{(t-1)}(\boldsymbol{\lambda}_k)\right\}$. Then we have the supremum norm

$$\begin{aligned}
&\left\|\boldsymbol{\Gamma}\left(\mathcal{M}^{(t-1)}\right)-\boldsymbol{\Gamma}\left(\widetilde{\mathcal{M}}^{(t-1)}\right)\right\|_{\sup}\\
&=\max_{k,j,n}\left|p_{k\to j}^{(t)}\left(\boldsymbol{\lambda}_k^{[n]}\right)-\tilde{p}_{k\to j}^{(t)}\left(\boldsymbol{\lambda}_k^{[n]}\right)\right|\\
&=\max_{k,j,n}\left|\prod_{i\in\mathcal{J}\setminus\{j\}}\sum_{h\in\mathcal{K}\setminus\{k\}}\sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}}\left(\exp\left(\mu F_i(\boldsymbol{\Lambda}_i)\right)\left(\prod_{q\in\mathcal{K}\setminus\{k\}}\right.\right.\right.\\
&\qquad\left.\left.\left.p_{q\to i}^{(t-1)}(\boldsymbol{\lambda}_q)-\prod_{q\in\mathcal{K}\setminus\{k\}}\tilde{p}_{q\to i}^{(t-1)}(\boldsymbol{\lambda}_q)\right)\right)\right|\\
&\overset{(a)}{\leq}\max_{j}\prod_{i\in\mathcal{J}\setminus\{j\}}\sum_{h\in\mathcal{K}\setminus\{k\}}\sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}}\\
&\qquad\left|\prod_{q\in\mathcal{K}\setminus\{k\}}p_{q\to i}^{(t-1)}(\boldsymbol{\lambda}_q)-\prod_{q\in\mathcal{K}\setminus\{k\}}\tilde{p}_{q\to i}^{(t-1)}(\boldsymbol{\lambda}_q)\right|\\
&\overset{(b)}{\leq}(K-1)N^{K-1}\max_{j}\\
&\qquad\prod_{i\in\mathcal{J}\setminus\{j\}}\max_{q\in\mathcal{K}\setminus\{k\},n}\left|p_{q\to i}^{(t-1)}\left(\boldsymbol{\lambda}_q^{[n]}\right)-\tilde{p}_{q\to i}^{(t-1)}\left(\boldsymbol{\lambda}_q^{[n]}\right)\right|\\
&\leq(K-1)N^{K-1}\max_{j,q\in\mathcal{K}\setminus\{k\},n}\left|p_{q\to i}^{(t-1)}\left(\boldsymbol{\lambda}_q^{[n]}\right)-\tilde{p}_{q\to i}^{(t-1)}\left(\boldsymbol{\lambda}_q^{[n]}\right)\right|^{J-1}\\
&\leq(K-1)N^{K-1}\max_{j,k,n}\left|p_{k\to i}^{(t-1)}\left(\boldsymbol{\lambda}_k^{[n]}\right)-\tilde{p}_{k\to i}^{(t-1)}\left(\boldsymbol{\lambda}_k^{[n]}\right)\right|\\
&=(K-1)N^{K-1}\left\|\mathcal{M}^{(t-1)}-\widetilde{\mathcal{M}}^{(t-1)}\right\|_{\sup}.
\end{aligned}\tag{28}$$

The inequality (a) in (28) is derived by exploiting the following two facts: 1) $0<\exp(\mu F_i(\boldsymbol{\Lambda}))\leq 1$, since $F_i(\boldsymbol{\Lambda})$ is non-positive and μ is positive, and 2) $\sum_s|x_s|\leq|\sum_s(x_s)|$ for arbitrary x_s. The inequality (b) in (28) can be obtained from: 1) the following lemma, and 2) the fact that $\sum_{h\in\mathcal{K}\setminus\{k\}}\sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}}$ has to carry out the additions of N^{K-1} items.

Lemma 2: Given $0\leq a_1,\cdots,a_K\leq 1$ and $0\leq\tilde{a}_1,\cdots,\tilde{a}_K\leq 1$, we have

$$\max_{k\in\mathcal{K}}\left|\prod_{q\in\mathcal{K}\setminus\{k\}}a_q-\prod_{q\in\mathcal{K}\setminus\{k\}}\tilde{a}_q\right|\leq(K-1)\max_{q\in\mathcal{K}\setminus\{k\}}|a_q-\tilde{a}_q|.\tag{29}$$

Proof: Please refer to Appendix F.

From (28), we can infer that $\boldsymbol{\Gamma}$ is a continuous mapping, since the coefficient $(K-1)N^{K-1}$ is a constant, and this completes the proof. ☐

Appendix B
Proof of Theorem 1

Let $\mathcal{S}$ be the collection of the message set $\mathcal{M}^{(t)}$. The mapping function $\boldsymbol{\Theta}$ maps $\mathcal{S}$ to $\mathcal{S}$ with the aid of the function $\boldsymbol{\Gamma}$. According to Lemma 1, $\boldsymbol{\Theta}$ is continuous since $\boldsymbol{\Gamma}$ is continuous. Furthermore, it is clear that the set $\mathcal{S}$ is convex, closed and bounded. Based on Schauder's fixed point theorem, $\boldsymbol{\Theta}$ has a fixed point. This completes the proof. ☐

Appendix C
Proof of Theorem 2

A. The Average Degree of Factor Nodes

Without a loss of generality, we carry out the analysis for a typical MU located at the origin and assume that the potential serving SBSs are located at the point x_B. The fading (power) is denoted by h_{x_B}, which is assumed to be exponentially distributed, i.e., we have $h_{x_B}\sim\exp(1)$. The path-loss function is given by $\|x_B\|^{-\alpha}$, where $\|\cdot\|$ denotes the Euclidian distance.

The average degree of a factor node in the factor graph is equivalent to the number of SBSs that can provide a high enough SINR ($\geq\delta$) for the typical MU, which can be formulated as

$$N_B=\int_{\mathbb{R}^2}\lambda_B\Pr\left(\rho(x_B)\geq\delta\right)\mathrm{d}x_B,\tag{30}$$

where $\rho(x_B)$ represents the SINR at the typical MU received from the SBSs located at x_B.

We first focus on the probability $\Pr(\rho(x_B)\geq\delta)$ in (30) as follows.

$$\begin{aligned}
\Pr\left(\rho(x_B)\geq\delta\right)&=\Pr\left(\frac{Ph_{x_B}\|x_B\|^{-\alpha}}{\sum\limits_{x_k\in\Phi_B}Ph_{x_k}\|x_k\|^{-\alpha}+\sigma^2}\geq\delta\right)\\
&=\Pr\left(h_{x_B}\geq\frac{\delta(I+\sigma^2)}{P\|x_B\|^{-\alpha}}\right)\\
&=\mathbb{E}_I\left(\exp(-sI)\right)\exp(-s\sigma^2),
\end{aligned}\tag{31}$$

where x_k denotes the location of an interfering SBS, $I \triangleq \sum_{x_k \in \Phi_B} P h_{x_k} \|x_k\|^{-\alpha}$ represents the aggregate interference, and $s = \frac{\delta \|x_B\|^{\alpha}}{P}$. The last step is due to the exponential distribution of h_{x_B}. Then, we derive $\mathbb{E}_I(\exp(-sI))$ in (31) as

$$\begin{aligned}
&\mathbb{E}_I\left(\exp(-sI)\right)\\
&\overset{(a)}{=} \mathbb{E}_{\Phi_B}\left(\prod_{x_k \in \Phi_B} \int_0^{\infty} \exp\left(-sPh_{x_k}\|x_k\|^{-\alpha}\right)\exp\left(-h_{x_k}\right)\mathrm{d}h_{x_k}\right)\\
&\overset{(b)}{=} \exp\left(-\lambda_B \int_{\mathbb{R}^2}\left(1-\frac{1}{1+sP\|x_k\|^{-\alpha}}\right)\mathrm{d}x_k\right)\\
&= \exp\left(-2\pi\lambda_B \frac{1}{\alpha}(sP)^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)\right), \qquad (32)
\end{aligned}$$

where (a) is based on the independence of channel fading, and (b) follows from $\mathbb{E}\left(\prod_x u(x)\right) = \exp(-\lambda \int_{\mathbb{R}^2}(1-u(x))\mathrm{d}x)$, where $x \in \Phi$ and Φ is an PPP in $\mathbb{R}^2$ with the intensity λ [30].

Based on the derivation above, the average degree of the typical MU can be calculated as

$$\begin{aligned}
N_B &= \lambda_B \int_{\mathbb{R}^2} \\
&\quad \exp\left(-2\pi\frac{\lambda_B}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)\|x_B\|^2 - \frac{\delta\sigma^2}{P}\|x_B\|^{\alpha}\right)\mathrm{d}x_B\\
&= 2\pi\lambda_B \int_0^{\infty} \exp\left(-2\pi\frac{\lambda_B}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) r^2 - \frac{\delta\sigma^2}{P} r^{\alpha}\right) r\mathrm{d}r. \qquad (33)
\end{aligned}$$

B. The Average Degree of Variable Nodes

In this subsection, we consider a typical SBS which is located at the origin, and assume that an MU is located at the point x_U. The average degree of a variable node in the factor graph is equivalent to the number of MUs that can receive at a high enough SINR ($\geq \delta$) from the typical SBS, which can be formulated as

$$N_U = \int_{\mathbb{R}^2} \lambda_U \Pr\left(\rho(x_U) \geq \delta\right)\mathrm{d}x_U, \qquad (34)$$

where $\rho(x_U)$ represents the received SINR at the MU located at x_U from the typical SBS, i.e.,

$$\begin{aligned}
&\Pr\left(\rho(x_U) \geq \delta\right)\\
&= \Pr\left(\frac{P h_{x_U}\|x_U\|^{-\alpha}}{\sum_{x_k \in \Phi_B} P h_{x_k}\|x_k - x_U\|^{-\alpha} + \sigma^2} \geq \delta\right), \qquad (35)
\end{aligned}$$

where x_k denotes the location of an interfering SBS.

Since the PPP is a stationary process, the distribution of $\|x_k - x_U\|$ is independent of the value of x_U, i.e., we have $p(\|x_k - x_U\|) = p(\|x_k\|)$, where $p(\cdot)$ represents the probability density function. Then, we have similar results to Eq. (31). That is, we have

$$\Pr\left(\rho(x_U) > \delta\right) = \mathbb{E}_I\left(\exp(-sI)\right)\exp(-s\sigma^2), \qquad (36)$$

where $s = \frac{\delta\|x_U\|^{\alpha}}{P}$. Then we arrive at

$$N_U = 2\pi\lambda_U \int_0^{\infty} \exp\left(-2\pi\frac{\lambda_B}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) r^2 - \frac{\delta\sigma^2}{P} r^{\alpha}\right) r\mathrm{d}r. \qquad (37)$$

By combining Eqs. (37) and (33), we complete the proof. □

APPENDIX D
PROOF OF COROLLARY 1

When ignoring the noise, we have

$$\begin{aligned}
&Z(\lambda_B, P, \alpha, \delta)\\
&= \int_0^{\infty} \exp\left(-\frac{2\pi\lambda_B}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) r^2\right) r\mathrm{d}r\\
&= \frac{1}{2}\int_0^{\infty} \exp\left(-\lambda_B \frac{2\pi}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) t\right)\mathrm{d}t\\
&= \frac{1}{2\lambda_B \frac{2\pi}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)} = \frac{\alpha}{4\pi\lambda_B B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)\delta^{\frac{2}{\alpha}}}. \qquad (38)
\end{aligned}$$

By substituting the above expression into (17) and (16), we obtain (20) and (21) respectively. This completes the proof. □

APPENDIX E
PROOF OF THEOREM 3

We conduct the analysis for a typical MU that is located at the origin. We assume that when downloading a file in $\mathcal{F}_n$, the MU will always associate with its nearest SBS, which caches $\mathcal{F}_n$. Note that the OP derived under this assumption is an upper bound for the exact OP. This is because the MU will associate with the second-nearest SBS if it can provide a higher received SINR than that provided by the nearest SBS. Therefore, in some cases, the nearest SBS cannot provide a higher enough SINR ($\geq \delta$), while the second-nearest SBS can. According to our assumption, we will neglect these cases, which leads to a higher OP.

Let us denote by z the distance between the typical MU and the nearest SBS that caches $\mathcal{F}_n$. The location of the nearest SBS caching $\mathcal{F}_n$ is denoted by x_Z. The fading (power) for an SBS located at x_B, $\forall x_B \in \Phi_B$, is denoted by h_{x_B}, which is assumed to be exponentially distributed, i.e., $h_{x_B} \sim \exp(1)$. The path-loss function for a given point x_B is $\|x_B\|^{-\alpha}$.

When random caching is adopted, the distribution of the SBSs that cache $\mathcal{F}_n$ can be modeled as an PPP with the intensity of $\Omega_{\mathcal{F}_n}\lambda_B$. The event that the typical MU can download a file in $\mathcal{F}_n$ from an SBS means that the received SINR from the nearest

SBS which caches $\mathcal{F}_n$ is no less than the threshold δ. Let us denote by $\rho(x_Z)$ the received SINR at the typical MU from the nearest SBS. Then the average probability that the MU can download the file from an SBS is

$$\begin{aligned}
&\Pr\left(\rho(x_Z) \geq \delta\right) \\
&= \int_0^\infty \Pr\left(\frac{h_{x_Z} z^{-\alpha}}{\sum\limits_{x_k \in \Phi_B \setminus \{x_Z\}} h_{x_k} \|x_k\|^{-\alpha}} \geq \delta \,\middle|\, z \right) f_Z(z) \mathrm{d}z \\
&= \int_0^\infty \Pr\left(h_{x_Z} \geq \frac{\delta \left(\sum\limits_{x_k \in \Phi_B \setminus \{x_Z\}} h_{x_k} \|x_k\|^{-\alpha} \right)}{z^{-\alpha}} \,\middle|\, z \right) \\
&\quad \cdot 2\pi \Omega_{\mathcal{F}_n} \lambda_B z \exp\left(-\pi \Omega_{\mathcal{F}_n} \lambda_B z^2\right) \mathrm{d}z \\
&= \int_0^\infty \mathbb{E}_I\left(\exp(-z^\alpha \delta I)\right) 2\pi \Omega_{\mathcal{F}_n} \lambda_B z \exp\left(-\pi \Omega_{\mathcal{F}_n} \lambda_B z^2\right) \mathrm{d}z,
\end{aligned} \tag{39}$$

where we have $I \triangleq \sum\limits_{x_k \in \Phi_B \setminus \{x_Z\}} h_{x_k} \|x_k\|^{-\alpha}$, and the PDF of z, i.e., $f_Z(z)$, is derived by the null probability of a Poisson process with the intensity of $\Omega_{\mathcal{F}_n} \lambda_B$. Note that the interference I consists of I_1 and I_2, where I_1 is emanating from the SBSs caching the FGs $\mathcal{F}_q, \forall q \in \boldsymbol{\mathcal{N}}, q \neq n$, while I_2 is from the SBSs caching $\mathcal{F}_n$ excluding x_Z. The SBSs contributing to I_1, denoted by $\Phi_{\bar{n}}$, have the intensity $(1 - \Omega_{\mathcal{F}_n})\lambda_B$, while those contributing to I_2, denoted by Φ_n, have the intensity $\Omega_{\mathcal{F}_n} \lambda_B$. Correspondingly, the calculation of $\mathbb{E}_I(\exp(-z^\alpha \delta I))$ will be split into the product of two expectations over I_1 and I_2. The expectation over I_1 directly follows (32), i.e., we have

$$\mathbb{E}_{I_1}\left(\exp(-z^\alpha \delta I_1)\right) = \exp\Big(-\pi \left(1 - \Omega_{\mathcal{F}_n}\right) \lambda_B C(\delta, \alpha) z^2\Big), \tag{40}$$

where $C(\delta, \alpha)$ has been defined as $\frac{2}{\alpha} \delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1 - \frac{2}{\alpha}\right)$. The expectation over I_2 has to take into account z as the distance from the nearest interfering SBS, i.e., we obtain

$$\begin{aligned}
&\mathbb{E}_{I_2}\left(\exp(-z^\alpha \delta I_2)\right) \\
&= \exp\left(-\Omega_{\mathcal{F}_n} \lambda_B 2\pi \int_z^\infty \left(1 - \frac{1}{1 + z^\alpha \delta r^{-\alpha}}\right) r \mathrm{d}r\right) \\
&\overset{(a)}{=} \exp\left(-\Omega_{\mathcal{F}_n} \lambda_B \pi \delta^{\frac{2}{\alpha}} z^2 \frac{2}{\alpha} \int_{\delta^{-1}}^\infty \frac{x^{\frac{2}{\alpha}-1}}{1+x} \mathrm{d}x\right) \\
&\overset{(b)}{=} \exp\left(-\Omega_{\mathcal{F}_n} \lambda_B \pi \delta z^2 \frac{2}{\alpha - 2} {}_2F_1\left(1, 1 - \frac{2}{\alpha}; 2 - \frac{2}{\alpha}; -\delta\right)\right),
\end{aligned} \tag{41}$$

where (a) defines $x \triangleq \delta^{-1} z^{-\alpha} r^\alpha$, and ${}_2F_1(\cdot)$ in (b) is the hypergeometric function. Since we have defined $A(\delta, \alpha) = \frac{2\delta}{\alpha-2} {}_2F_1\left(1, 1 - \frac{2}{\alpha}; 2 - \frac{2}{\alpha}; -\delta\right)$, by substituting (40) and (41) into (39), we have

$$\begin{aligned}
&\Pr\left(\rho(x_Z) \geq \delta\right) = \int_0^\infty \exp\left(-\pi \left(1 - \Omega_{\mathcal{F}_n}\right) \lambda_B C(\delta, \alpha) z^2\right) \\
&\exp\left(-\pi \Omega_{\mathcal{F}_n} \lambda_B z^2 A(\delta, \alpha)\right) 2\pi \Omega_{\mathcal{F}_n} \lambda_B z \exp\left(-\pi \Omega_{\mathcal{F}_n} \lambda_B z^2\right) \mathrm{d}z \\
&= \frac{\Omega_{\mathcal{F}_n}}{C(\delta, \alpha)\left(1 - \Omega_{\mathcal{F}_n}\right) + A(\delta, \alpha) \Omega_{\mathcal{F}_n} + \Omega_{\mathcal{F}_n}}.
\end{aligned} \tag{42}$$

It is clear that $\Pr(\mathcal{Q}_n) = 1 - \Pr(\rho(z) \geq \delta)$. This completes the proof. □

Appendix F
Proof of Lemma 2

Without loss of generality, we assume $k = 1$. Then (29) becomes

$$\left| \prod_{q=2}^K a_q - \prod_{q=2}^K \tilde{a}_q \right| \leq (K - 1) \max_{q \in \{2, \cdots, K\}} |a_q - \tilde{a}_q|. \tag{43}$$

Again, without loss of generality, we assume

$$|a_2 - \tilde{a}_2| \geq \cdots \geq |a_K - \tilde{a}_K|. \tag{44}$$

First, we prove that $|a_{K-1} a_K - \tilde{a}_{K-1} \tilde{a}_K| \leq 2|a_{K-1} - \tilde{a}_{K-1}|$, under the condition of $|a_{K-1} - \tilde{a}_{K-1}| \geq |a_K - \tilde{a}_K|$. To prove this, we discuss the following possible cases.

1) When $a_{K-1} \geq \tilde{a}_{K-1}$ and $a_K \geq \tilde{a}_K$: We have $a_K \leq a_{K-1} - \tilde{a}_{K-1} + \tilde{a}_K$. Then

$$\begin{aligned}
&|a_{K-1} a_K - \tilde{a}_{K-1} \tilde{a}_K| \\
&\quad \leq |a_{K-1}(a_{K-1} - \tilde{a}_{K-1} + \tilde{a}_K) - \tilde{a}_{K-1} \tilde{a}_K| \\
&\quad = |(a_{K-1} + \tilde{a}_K)(a_{K-1} - \tilde{a}_{K-1})| \\
&\quad \leq 2|a_{K-1} - \tilde{a}_{K-1}|.
\end{aligned} \tag{45}$$

2) When $a_{K-1} \geq \tilde{a}_{K-1}$, $a_K \leq \tilde{a}_K$, and $a_{K-1} a_K \geq \tilde{a}_{K-1} \tilde{a}_K$: We have

$$\begin{aligned}
|a_{K-1} a_K - \tilde{a}_{K-1} \tilde{a}_K| &\leq |a_{K-1} \tilde{a}_K - \tilde{a}_{K-1} \tilde{a}_K| \\
&= |a_{K-1} - \tilde{a}_{K-1}| \tilde{a}_K \leq |a_{K-1} - \tilde{a}_{K-1}|.
\end{aligned} \tag{46}$$

3) When $a_{K-1} \geq \tilde{a}_{K-1}$, $a_K \leq \tilde{a}_K$, and $a_{K-1} a_K \leq \tilde{a}_{K-1} \tilde{a}_K$: We have

$$\begin{aligned}
|\tilde{a}_{K-1} \tilde{a}_K - a_{K-1} a_K| &\leq |a_{K-1} \tilde{a}_K - a_{K-1} a_K| \\
&= |a_K - \tilde{a}_K| a_{K-1} \leq |a_{K-1} - \tilde{a}_{K-1}|.
\end{aligned} \tag{47}$$

4) When $a_{K-1} \leq \tilde{a}_{K-1}$, $a_K \geq \tilde{a}_K$, and $a_{K-1} a_K \geq \tilde{a}_{K-1} \tilde{a}_K$: We have

$$\begin{aligned}
|a_{K-1} a_K - \tilde{a}_{K-1} \tilde{a}_K| &\leq |\tilde{a}_{K-1} a_K - \tilde{a}_{K-1} \tilde{a}_K| \\
&= |a_K - \tilde{a}_K| \tilde{a}_{K-1} \leq |a_{K-1} - \tilde{a}_{K-1}|.
\end{aligned} \tag{48}$$

5) When $a_{K-1} \le \tilde{a}_{K-1}$, $a_K \ge \tilde{a}_K$, and $a_{K-1}a_K \le \tilde{a}_{K-1}\tilde{a}_K$: We have

$$\begin{aligned}|\tilde{a}_{K-1}\tilde{a}_K - a_{K-1}a_K| &\le |\tilde{a}_{K-1}a_K - a_{K-1}a_K| \\ &= |a_{K-1} - \tilde{a}_{K-1}|a_K \le |a_{K-1} - \tilde{a}_{K-1}|. \end{aligned} \tag{49}$$

6) When $a_{K-1} \le \tilde{a}_{K-1}$, $a_K \le \tilde{a}_K$: We have $a_K \ge \tilde{a}_K + a_{K-1} - \tilde{a}_{K-1}$. Then

$$\begin{aligned}|\tilde{a}_{K-1}\tilde{a}_K - a_{K-1}a_K| &\le |\tilde{a}_{K-1}\tilde{a}_K - a_{K-1}(\tilde{a}_K + a_{K-1} - \tilde{a}_{K-1})| \\ &= |(a_{K-1} + \tilde{a}_K)(\tilde{a}_{K-1} - a_{K-1})| \\ &\le 2|a_{K-1} - \tilde{a}_{K-1}|. \end{aligned} \tag{50}$$

From the above discussions, we can see that $|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| \le 2|a_{K-1} - \tilde{a}_{K-1}|$.

Second, as there is $|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| \le 2|a_{K-1} - \tilde{a}_{K-1}|$, we have $|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| \le 2|a_{K-2} - \tilde{a}_{K-2}|$. With this condition, we can prove that $|a_{K-2}a_{K-1}a_K - \tilde{a}_{K-2}\tilde{a}_{K-1}\tilde{a}_K| \le 3|a_{K-2} - \tilde{a}_{K-2}|$ by following the similar steps above. By doing this iteratively, we have

$$\left|\prod_{q=2}^{K} a_q - \prod_{q=2}^{K} \tilde{a}_q\right| \le (K-1)|a_2 - \tilde{a}_2|. \tag{51}$$

This completes the proof. □

References

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2013–2018," Cisco, San Jose, CA, USA. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf

[2] F. Boccardi *et al.*, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[3] A. Damnjanovic *et al.*, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 10–21, Jun. 2011.

[4] J. Akhtman and L. Hanzo, "Heterogeneous networking: An enabling paradigm for ubiquitous wireless communications," *Proc. IEEE*, vol. 98, no. 2, pp. 135–138, Feb. 2010.

[5] S. Bayat, R. Louie, Z. Han, B. Vucetic, and Y. Li, "Distributed user association and femtocell allocation in heterogeneous wireless networks," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 3027–3043, Aug. 2014.

[6] M. Mirahmadi, A. Al-Dweik, and A. Shami, "Interference modeling and performance evaluation of heterogeneous cellular networks," *IEEE Trans. Commun.*, vol. 62, no. 6, pp. 2132–2144, Jun. 2014.

[7] A. Gupta, H. Dhillon, S. Vishwanath, and J. Andrews, "Downlink multi-antenna heterogeneous cellular network with load balancing," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 4052–4067, Nov. 2014.

[8] Y. Kishiyama, A. Benjebbour, T. Nakamura, and H. Ishii, "Future steps of LTE-A: Evolution toward integration of local area and wide area systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 12–18, Feb. 2013.

[9] T. Nakamura *et al.*, "Trends in small cell enhancements in LTE Advanced," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 98–105, Feb. 2013.

[10] Y. Li, H. Celebi, M. Daneshmand, C. Wang, and W. Zhao, "Energy-efficient femtocell networks: Challenges and opportunities," *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 99–105, Dec. 2013.

[11] N. Golrezaei, A. Molisch, A. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[12] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen, "Coding or not: Optimal mobile data offloading in opportunistic vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 318–333, Feb. 2014.

[13] J. Xu, Q. Hu, W.-C. Lee, and D. Lee, "Performance evaluation of an optimal cache replacement policy for wireless data dissemination," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 125–139, Jan. 2004.

[14] Y. Li *et al.*, "Multiple mobile data offloading through disruption tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1579–1596, Jul. 2014.

[15] D. Chambers, "Data caching reduces backhaul costs for small cells and Wi-Fi," Thinksmallcell, Bath, U.K., Thinksmallcell Forum, Tech. Rep., May 2013.

[16] H. Sarkissian, "The business case for caching in 4G LTE networks," Wireless 2020, Tech. Rep., 11 2014.

[17] "Rethinking the small cell business model," Intel, Santa Clara, CA, USA, Tech. Rep., 2012.

[18] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. 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.

[19] N. Golrezaei, P. Mansourifard, A. Molisch, and A. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[20] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," arXiv preprint arXiv:1305.5216, to be published. AQ1

[21] M. Ji, G. Caire, and A. Molisch, "Optimal throughput-outage trade-off in wireless one-hop caching networks," in *Proc. IEEE ISIT*, Jul. 2013, pp. 1461–1465.

[22] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[23] K. Shanmugam, N. Golrezaei, A. Dimakis, A. 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.

[24] C. C. Moallemi and B. Van Roy, "Resource allocation via message passing," *INFORMS J. Comput.*, vol. 23, no. 2, pp. 205–219, 2011.

[25] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum–product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[26] S. Bavarian and J. Cavers, "Reduced-complexity belief propagation for system-wide MUD in the uplink of cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 3, pp. 541–549, Apr. 2008.

[27] I. Sohn, S. H. Lee, and J. Andrews, "Belief propagation for distributed downlink beamforming in cooperative MIMO cellular networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 12, pp. 4140–4149, Dec. 2011.

[28] D. Stoyan, W. Kendall, and M. Mecke, *Stochastic Geometry and Its Applications*, 2nd ed. New York, NY, USA: Wiley, 2003.

[29] M. Haenggi, J. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1029–1046, Sep. 2009.

[30] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes, Volume I: Elementary Theory and Methods*. New York, NY, USA: Springer-Verlag, 1996.

[31] H. Dhillon, R. Ganti, F. Baccelli, and J. 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.

[32] S. Rangan and R. Madan, "Belief propagation methods for intercell interference coordination in femtocell networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 631–640, Apr. 2012.

[33] H.-S. Jo, Y. J. Sang, P. Xia, and J. Andrews, "Heterogeneous cellular networks with flexible cell association: A comprehensive downlink SINR analysis," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3484–3495, Oct. 2012.

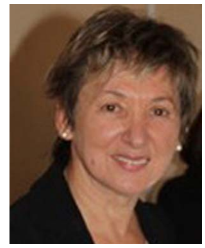
[34] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, 2007, pp. 1–14.

Jun Li (M'09) received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009. From January 2009 to June 2009, he was with the Department of Research and Innovation, Alcatel Lucent Shanghai Bell, as a Research Scientist. From June 2009 to April 2012, he was a Postdoctoral Fellow at the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia. Since April 2012, he has been a Research Fellow at the School of Electrical Engineering, The University of Sydney, Sydney, Australia. His research interests include network information theory, channel coding theory, wireless network coding, and cooperative communications. He has served as the Technical Program Committee Member for several international conferences such as GlobeCom2015, ICC2014, VTC2014 (Fall), and ICC2014.

Youjia Chen received the B.S. and M.S degrees in communication engineering from Nanjing University, Nanjing, China, in 2005 and 2008, respectively. She is currently working toward the Ph.D. degree in wireless engineering at The University of Sydney, Sydney, Australia. Her current research interests include resource management, load balancing, and caching strategy in heterogeneous cellular networks.

Zihuai Lin (S'98–M'99–SM'11) received the Ph.D. degree in electrical engineering from Chalmers University of Technology, Gothenburg, Sweden, in 2006. Prior to this, he has held positions at Ericsson Research, Stockholm, Sweden. Following Ph.D. graduation, he was a Research Associate Professor at Aalborg University, Aalborg, Denmark, and currently at the School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia. His research interests include graph theory, source/channel/network coding, coded modulation, MIMO, OFDMA, SC-FDMA, radio resource management, cooperative communications, small-cell networks, 5G cellular systems, etc.

Wen Chen (M'03–SM'11) received the B.S. and M.S. degrees from Wuhan University, Wuhan, China, in 1990 and 1993, respectively, and the Ph.D. degree from the University of Electro-Communications, Tokyo, Japan, in 1999. He was a Researcher at the Japan Society for the Promotion of Science from 1999 through 2001. In 2001, he joined the University of Alberta, Canada, starting as a Postdoctoral Fellow with the Information Research Laboratory and continuing as a Research Associate in the Department of Electrical and Computer Engineering. Since 2006, he has been a Full Professor at the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is also the Director of the Institute for Signal Processing and Systems. His interests cover network coding, cooperative communications, cognitive radio, and MIMO-OFDM systems.



Branka Vucetic (M'83–SM'00–F'03) currently holds the Peter Nicol Russel Chair of Telecommunications Engineering at The University of Sydney, Sydney, Australia. During her career, she has held various research and academic positions in Yugoslavia, Australia, U.K., and China. She has co-authored four books and more than 400 papers in telecommunications journals and conference proceedings. Her research interests include wireless communications, coding, digital communication theory, and machine-to-machine communications. Prof. Vucetic has been elected to the grade of IEEE Fellow for contributions to the theory and applications of channel coding.

Lajos Hanzo (M'91–SM'92–F'04) received the M.S. degree in electronics and the Ph.D. degree from the Technical University of Budapest, Budapest, Hungary, in 1976 and 1983, respectively; the D.Sc. degree from the University of Southampton, Southampton, U.K., in 2004; and the "Doctor Honoris Causa" degree from the Technical University of Budapest in 2009. During his 38-year career in telecommunications, he has held various research and academic posts in Hungary, Germany, and the U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, where he holds the Chair in Telecommunications. He is currently directing a 60-strong academic research team, working on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council, the European Research Council's Advanced Fellow Grant, and the Royal Society's Wolfson Research Merit Award. During 2008–2012, he was a Chaired Professor at Tsinghua University, Beijing. He is an enthusiastic supporter of industrial and academic liaison and offers a range of industrial courses. He has successfully supervised about 100 Ph.D. students, coauthored 20 John Wiley/IEEE Press books on mobile radio communications totaling in excess of 10 000 pages, and published more than 1500 research entries at IEEE Xplore. His research is funded by the European Research Council's Senior Research Fellow Grant. Dr. Hanzo is a Fellow of the Royal Academy of Engineering, the Institution of Engineering and Technology, and the European Association for Signal Processing. He is also a Governor of the IEEE Vehicular Technology Society. During 2008–2012, he was the Editor-in-Chief of IEEE Press. He has served as the Technical Program Committee Chair and the General Chair of IEEE conferences, has presented keynote lectures, and has been awarded a number of distinctions. He has more than 22 000 citations. For further information on research in progress and associated publications, please refer to http://www-mobile.ecs.soton.ac.uk

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

AQ1 = Please provide publication update in Ref. [20].
AQ2 = Please provide details on the educational background of author Branka Vucetic.

END OF ALL QUERIES

Distributed Caching for Data Dissemination in the Downlink of Heterogeneous Networks

Jun Li, *Member, IEEE*, Youjia Chen, Zihuai Lin, *Senior Member, IEEE*, Wen Chen, *Senior Member, IEEE*, Branka Vucetic, *Fellow, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—Heterogeneous cellular networks (HCNs) with embedded small cells are considered, where multiple mobile users wish to download network content of different popularity. By caching data into the small-cell base stations, we will design distributed caching optimization algorithms via belief propagation (BP) for minimizing the downloading latency. First, we derive the delay-minimization objective function and formulate an optimization problem. Then, we develop a framework for modeling the underlying HCN topology with the aid of a factor graph. Furthermore, a distributed BP algorithm is proposed based on the network's factor graph. Next, we prove that a fixed point of convergence exists for our distributed BP algorithm. In order to reduce the complexity of the BP, we propose a heuristic BP algorithm. Furthermore, we evaluate the average downloading performance of our HCN for different numbers and locations of the base stations and mobile users, with the aid of stochastic geometry theory. By modeling the nodes distributions using a Poisson point process, we develop the expressions of the average factor graph degree distribution, as well as an upper bound of the outage probability for random caching schemes. We also improve the performance of random caching. Our simulations show that 1) the proposed distributed BP algorithm has a near-optimal delay performance, approaching that of the high-complexity exhaustive search method; 2) the modified BP offers a good delay performance at low communication complexity; 3) both the average degree distribution and the outage upper bound analysis relying on stochastic geometry match well with our Monte-Carlo simulations; and 4) the optimization based on the upper bound provides both a better outage and a better delay performance than the benchmarks.**

Manuscript received December 24, 2014; revised May 9, 2015 and July 4, 2015; accepted July 7, 2015. This work was supported in part by Australian Research Council Programs under Grant DP120100405, by the National 973 Project under Grant 2012CB316106, by the NSF China under Grants 61328101, 61271230, and 61472190, by the STCSM Science and Technology Innovation Program under Grant 13510711200, and by the SEU National Key Laboratory on Mobile Communications under Grants 2013D11 and 2013D02. The associate editor coordinating the review of this paper and approving it for publication was Z. Dawy.

J. Li is with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: jleesr80@gmail.com).

Y. Chen, Z. Lin, and B. Vucetic are with the School of Electrical and Information Engineering, The University of Sydney, Sydney, N.S.W. 2006, Australia (e-mail: youjia.chen@sydney.edu.au; zihuai.lin@sydney.edu.au; branka.vucetic@sydney.edu.au).

W. Chen is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wenchen@sjtu.edu.cn).

L. Hanzo is with Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: lh@ecs.soton.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/TCOMM.2015.2455500

***Index Terms*—Wireless caching, heterogeneous cellular networks, belief propagation, stochastic geometry.**

I. INTRODUCTION

WIRELESS data traffic is expected to increase by a factor of 40 over the next five years, from the current level of 93 Petabytes to 3600 Petabytes per month [1], driven by a rapid increase in the number of mobile users (MU) and aggravated by their bandwidth-hungry mobile applications. A promising approach to enhancing the network capacity is to embed small cells relying on low-power base stations (BS) into the existing macro-cell based networks. These networks, which are referred to as heterogeneous cellular networks (HCN) [2]–[7], typically contain regularly deployed macro-cells and embedded femtocells as well as pico-cells [8]–[10] that are served by macrocell BSs (MBS) and small-cell BSs (SBS), respectively. The aim of these flexibly deployed low-power SBSs is to eliminate the coverage holes and to increase the capacity in hot-spots.

There is evidence that the MUs' downloading of video on-demand files is the main reason for the growth of data traffic over cellular networks [11]. According to the prediction of Cisco on mobile data traffic, the mobile video streaming traffic will occupy 72% percentage of the overall mobile data traffic by 2019. Often, there are numerous repetitive downloading requests of popular contents, such as online blockbusters, leading to redundant data streaming. The redundancy of data transmissions can be reduced by locally storing popular data, known as caching, into the local SBSs, effectively forming a local cloud caching system (LCCS). The LCCS brings the content closer to the MUs and alleviates redundant data transmissions via redirecting the downloading requests to local SBSs. Also, the SBSs are willing to cache files into their buffers as long as they can, since caching is capable of significantly reducing the tele-traffic load on their back-haul channels, which are expensive.

In [12], the authors study the caching strategies of delay-tolerant vehicular networks, where the data subscribers and "helpers" are always moving and the links between them are opportunistic. By proposing an efficient algorithm to carefully allocate the network resources to mobile data, the decision is made as to which content should use the erasure coding, as well as conceiving the coding policy for each mobile data. In [13], optimal cache replacement policies are investigated. The cache replacement process takes place after the data caching process has been completed, and determines which particular data item should be deleted from the cache, when the available storage space is insufficient for accommodating an item to be cached.

Since the HCN structure has been widely adopted in current cellular networks and will prevail in near-future networks, we are interested in the SBS-based LCCS in the context of HCNs. In contrast to the vehicular networks discussed in [12], [14], where the mobility and the opportunistic communication contact are important issues, in the context of HCNs, the BSs are always fixed, and the MUs are assumed to be moving at a low speed. Thus, we ignore the mobility issues in the HCNs and assume that each MU is associated with a fixed BS during file-downloading. At the time of writing, there are already technical reports highlighting the advantages of caching in HCNs [15]–[17]. Based on these reports, the LCCS with SBS caching for HCNs is capable of efficiently 1) reducing the transmission latency due to short distance between the SBSs and the MUs, 2) offloading redundant data streams from MBSs, and 3) alleviating heavy burdens on the back-haul channels of the SBSs. Therefore, SBS-based caching will bring about significant breakthroughs for future HCNs.

The concept of caching is common in wireline networks and computer systems. However, research on efficient caching design for wireless cellular networks relying on small cells is still in its infancy [11], [18]. Usually, data caching consists of two phases: data placement and data transmission. During the data placement phase, data is cached into local SBSs in order to form an LCCS. In the data transmission phase, MUs request data from the LCCS. The focus of wireless caching research is mainly on the optimization of data placement for ensuring that the downloading latency is minimized. The caching optimization is a non-trivial problem. This is due to the massive scale of video contents to be stored in the limited memory of the SBSs.

The survey papers [11], [18] report on a range of attractive caching architectures conceived for future cellular networks. In [19], a caching scheme is proposed for a device-to-device (D2D) based cellular network on the MUs' caching of popular data. In this scheme, the D2D cluster size was optimized for reducing the downloading delay. In [20], [21], the authors propose a caching scheme for wireless sensor networks, where the protocol model of [22] is adopted. In [23], a femto-caching scheme is proposed for a cellular network combined with SBSs, where the data placement at the SBSs is optimized in a centralized manner for reducing the transmission delay imposed. However, [23] considers an idealized system, where neither the interference nor the impact of wireless channels is taken into account. The associations between the MUs and the SBSs are pre-determined without considering the specific channel conditions encountered. Furthermore, this centralized optimization method assumes that the MBS has perfect knowledge of all the channel state information (CSI) between the MUs and SBSs, which is impractical.

Against this background, in this paper, we consider distributed caching solutions for HCNs operating under more practical considerations. Our contributions consist of two parts.

1) In the first part, we propose distributed caching algorithms for enhancing the downloading performance via belief propagation (BP) [24]. The BP algorithm is capable of decomposing a global optimization problem into multiple sub-problems, thereby offering an efficient distributive approach of solving the global optimization problem [25]–[27]. As the BP method has been widely adopted for distributively solving resource allocation in cellular networks, we arrange file placement via BP algorithms by viewing files as a type of resource.
2) In the second part, we analyze the average caching performance based on stochastic geometry theory [28], [29]. We are interested in optimizing the average performance of a set of HCNs, where the channels exhibit Rayleigh fading and the distributions of network nodes obey a Poisson point process (PPP) [30].

Specifically, our contributions in the first part are follows.

1) We commence by deriving the delay as our optimization objective function (OF) and formulate the problem as optimizing the file placement.
2) We develop a framework for modeling the associated factor graph based on the topology of the network. A distributed BP algorithm is proposed based on the factor graph, which allows the file placement to be optimized in a distributed manner between the MUs and SBSs.
3) We prove that a fixed point exists in the proposed BP algorithm and show that the BP algorithm is capable of converging to this fixed point under certain conditions.
4) To reduce the communication complexity, we propose a heuristic BP algorithm.

Our contributions in the second part are follows.

1) By following the stochastic geometry framework, we model the MUs and SBSs in the HCN as different ties of a PPP. Furthermore, we develop the average degree distribution of the factor graph in the BP algorithm.
2) A random caching scheme is proposed, where each SBS will cache a file with a pre-determined probability. We can characterize the average downloading performance by outage probability (OP) and develop a tight upper bound of the OP expression with a closed form under the random caching scheme.
3) Based on the upper bound derived, we further improve the OP performance of random caching by optimizing the probabilities for caching different files.

In the simulations, we first investigate the average degree distribution of the factor graph, as well as the OP and the delay of the random caching schemes, in conjunction with various PPP parameters and power settings. It is shown that both the degree distribution and our upper bound analysis match well with the results of Monte-Carlo simulations. Furthermore, the optimization based on the upper bound provides both a better OP and a better delay than the benchmarks. Then we evaluate the distributed BP algorithm in our HCNs having a fixed number of BSs and MUs. It is shown that the proposed distributed BP algorithm has a near-optimal performance, approaching that of the exhaustive search method. The heuristic BP also offers a relatively good performance, despite its significantly reduced communication complexity.

The rest of this paper is organized as follows. We describe the system model in Section II and present the distributed file downloading problem relying on caching in Section III. We then propose a distributed BP algorithm in Section IV, where the proof of existence for a fixed point is also presented. In Section V, a heuristic BP algorithm is proposed for reducing the associated communication complexity. Our stochastic geometry based analysis is detailed in Section VI, where the average degree distribution of the factor graph and the OP of the random caching scheme are developed. Our simulation results are summarized in Section VII, while our conclusions are provided in Section VIII.

II. System Model

Let us consider an HCN consisting of a single MBS and K SBSs illuminating both femto-cells and pico-cells, while supporting J MUs randomly located in the network. Let us denote by $\mathcal{B}_0$ the MBS and by $\boldsymbol{\mathcal{B}} = \{\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_K\}$ the set of the SBSs, where $\mathcal{B}_k$, $k \in \boldsymbol{\mathcal{K}} = \{1, 2, \cdots, K\}$, represents the k-th SBS. Furthermore, denote by $\boldsymbol{\mathcal{U}} = \{\mathcal{U}_1, \mathcal{U}_2, \cdots \mathcal{U}_J\}$ the set of the MUs, where $\mathcal{U}_j$, $j \in \boldsymbol{\mathcal{J}} = \{1, 2, \cdots, J\}$, represents the j-th MU. The MBS $\mathcal{B}_0$ caches files into the memories of the SBSs during off-peak time via back-haul channels. Once the caching process is completed, the MBSs and SBSs are ready to act upon the downloading requests of the MUs.

We assume that a dedicated frequency band of bandwidth W is allocated to the downlink channels spanning from the SBSs to the MUs for file-dissemination. For reasons of careful load balancing, we consider the "SBS-first" constraint, where each MU will try to download data from its adjacent SBSs, unless the required files cannot be found in these SBSs. In this case, the MU will turn to the MBS for retrieving the required files. For the sake of simplicity, we assume that the MBS will support a fixed download rate, denoted by C_0, for the MUs in the channels which are orthogonal to those spanning from the SBSs to MUs.

In order to satisfy the "SBS-first" constraint for offloading data from the MBS, some incentives may be provided for the MUs. For example, downloading from the SBSs is much cheaper than from the MBS. Here, we assume that the download rate C_0 supported by the MBS is never higher than the lowest download rate supported by the SBSs. This limit imposed on the download rate from the MBS will not only encourage the MUs to download from the SBSs first, but also effectively control the data traffic of the MBS imposed by file downloading.

Denote by P_k the transmission power of the k-th SBS, and by σ^2 the noise power at each MU. The path-loss between $\mathcal{B}_k$ and the MU $\mathcal{U}_j$ is modeled as $d_{k,j}^{-\alpha}$, where $d_{k,j}$ is the distance between $\mathcal{B}_k$ and $\mathcal{U}_j$, and α is the path-loss exponent. The random channel between $\mathcal{B}_k$ and $\mathcal{U}_j$ is Rayleigh fading, whose coefficient $h_{k,j}$ has the average power of one. We assume that all the downlink channels spanning from the SBSs to the MUs are independent and identically distributed (i.i.d.).

Suppose that each file is split into multiple chunks and each chunk can be downloaded by an MU in a short time slot. Due to the short downloading time of a chunk, we assume furthermore that the probability of having two MUs streaming a chunk at the same time (or within a relative delay of a few seconds) from the same SBS is basically zero [20]. Hence, neither direct multicasting by exploiting the broadcast nature of the wireless medium nor network coding is considered. Furthermore, we focus our attention on the saturated scenario, where the SBSs keep transmitting data to the MUs [31]. Hence, each MU is subject to the interference imposed by all the other SBSs in $\boldsymbol{\mathcal{B}}$, when downloading files from its associated SBS. Given a channel realization $\mathbf{h}_j = [h_{1,j}, \cdots, h_{K,j}]$, the channel capacity between $\mathcal{B}_k$ and $\mathcal{U}_j$ can be calculated based on the signal-to-interference-plus-noise ratio (SINR) as

$$C_{k,j} = W \log \left(1 + \frac{h_{k,j}^2 d_{k,j}^{-\alpha} P_k}{\sum\limits_{q \in \boldsymbol{\mathcal{K}} \setminus \{k\}} h_{q,j}^2 d_{q,j}^{-\alpha} P_q + \sigma^2} \right). \quad (1)$$

Due to the 'SBS-first' constraint, we have $C_0 \leq C_{k,j}$, $\forall k \in \boldsymbol{\mathcal{K}}, j \in \boldsymbol{\mathcal{J}}$.

Denote by $\boldsymbol{\mathcal{F}}$ the library or set of files, which consists of Q popular files to be requested frequently by the MUs. The popularity distribution among the set $\boldsymbol{\mathcal{F}}$ is represented by $\boldsymbol{\mathcal{P}} = \{p_1, p_2, \cdots, p_Q\}$, where the MUs make independent requests of the f-th file, $f = 1, \cdots, Q$, with the probability of p_f. Without any loss of generality, all these files have the same size of M bits. We assume that $\mathcal{B}_0$ has a sufficiently large memory and hence accommodates the entire library of files, while the storage of each SBS is limited to G files, where we have $G < Q$.

Without a loss of generality, we assume that Q/G is an integer. The Q files in $\boldsymbol{\mathcal{F}}$ are divided into $N = Q/G$ file groups (FG), with each FG containing G files. The f-th file, $\forall f \in \{(n-1)G + 1, \cdots, nG\}$, is included in the n-th FG, $n \in \boldsymbol{\mathcal{N}} = \{1, \cdots, N\}$. We denote by $\mathcal{F}_n$ the n-th FG, and by $P_{\mathcal{F}_n}$ the probability that the MUs request a file in $\mathcal{F}_n$. Based on $\boldsymbol{\mathcal{P}}$, we have

$$P_{\mathcal{F}_n} = \sum_{f=(n-1)G+1}^{nG} p_f. \quad (2)$$

File caching is then carried out on the basis of FG, i.e., each SBS caches one of the N FGs.

III. Distributed File Downloading Relying on Caching

The caching-based distributed file downloading protocol consists of two stages. The first stage, or file placement stage, includes file content broadcasting and caching. In this stage, $\mathcal{B}_0$ broadcasts the FGs to the SBSs via the back-haul during off-peak periods. At the same time, the SBSs listen to the broadcasting from $\mathcal{B}_0$, and cache the FGs needed. The second stage, or file downloading stage, includes MU-SBS associations and file content transmissions. In this stage, each MU makes decisions as to which SBSs it should be associated with, and then starts to download files from the associated SBSs. When the requested files are not found in the adjacent SBSs, the MUs will turn to the MBS for these files.

A. File Placement Matrix

For assigning the N FGs to the K SBSs, we set up a file placement matrix $\boldsymbol{\Lambda}$ of size $K \times N$. The entry $\lambda_{k,n} \in \{0, 1\}$ in $\boldsymbol{\Lambda}$ indicates whether $\mathcal{F}_n$ is cached by $\mathcal{B}_k$ or not. We have $\lambda_{k,n} = 1$ if $\mathcal{F}_n$ is cached by $\mathcal{B}_k$, while $\lambda_{k,n} = 0$ otherwise. The

k-th row of $\mathbf{\Lambda}$ indicates which FG is cached by $\mathcal{B}_k$, and the n-th column indicates which BS caches $\mathcal{F}_n$. The number of the SBSs which cache $\mathcal{F}_n$ can be calculated as $\sum_{k\in\mathcal{K}}\lambda_{k,n}$. Since each SBS caches one FG, we have $\sum_{n\in\mathcal{N}}\lambda_{k,n}=1$.

B. MU-SBS Association

Denote by $\mathcal{H}(j)$ the subscript set of the specific SBSs, which are capable of providing a sufficiently high SINR for the MU $\mathcal{U}_j$. The SBSs in $\mathcal{H}(j)$ are the candidates for $\mathcal{U}_j$ to be potentially associated with. By setting an SINR threshold δ, $\mathcal{B}_k$ will be included in $\mathcal{H}(j)$ if and only if

$$\frac{h_{k,j}^2 d_{k,j}^{-\alpha} P_k}{\sum\limits_{q\in\mathcal{K}\setminus\{k\}} h_{q,j}^2 d_{q,j}^{-\alpha} P_q + \sigma^2} \geq \delta. \tag{3}$$

When requesting a file in $\mathcal{F}_n$, $\mathcal{U}_j$ first communicates with one of the SBSs in $\mathcal{H}(j)$ which caches $\mathcal{F}_n$. It is possible that more than one SBS in $\mathcal{H}(j)$ caches $\mathcal{F}_n$. In this case, $\mathcal{U}_j$ will associates with the optimal SBS, which imposes the minimum downloading delay.

It is clear that the downloading delay is inversely proportional to the downlink transmission rate. According to the file request assumption stipulated in the previous section, there is only a single MU connected to an SBS at each time. Thus, the maximum transmission rate from $\mathcal{B}_h$ to $\mathcal{U}_j$, $\forall\, h\in\mathcal{H}(j)$, is the channel capacity between them, i.e., $C_{h,j}$. When $\mathcal{U}_j$ tries to download a file in $\mathcal{F}_n$, it follows the maximum-capacity association criterion. Hence, $\mathcal{U}_j$ associates with $B_{\hat{h}}$ such that

$$\hat{h} = \underset{h\in\mathcal{H}(j)}{\arg\max}\left\{\lambda_{h,n}C_{h,j}\right\}. \tag{4}$$

When none of the SBSs in $\mathcal{H}(j)$ caches $\mathcal{F}_n$, i.e., we have $\lambda_{h,n}=0$, $\forall\, h\in\mathcal{H}(j)$, $\mathcal{U}_j$ will associate with the MBS for the requested file.

C. Optimization Problem Formulation

We now optimize the matrix $\mathbf{\Lambda}$ for minimizing the average delay of downloading a file. Only when the optimal $\mathbf{\Lambda}$ has been determined will the file-placement stage commence, where the files are placed according this optimal matrix. Once the MU-SBS associations have been determined, we can optimize the matrix $\mathbf{\Lambda}$ for minimizing the average delay of downloading a file. First, given the channel coefficients and the specific location of $\mathcal{U}_j$, the delay of downloading a file in $\mathcal{F}_n$ by $\mathcal{U}_j$ can be calculated as

$$D_{j,n} = \begin{cases} \frac{M}{\max_{h\in\mathcal{H}(j)}\{\lambda_{h,n}C_{h,j}\}}, & \exists\lambda_{h,n}\neq 0, \quad \forall\, h\in\mathcal{H}(j) \\ \frac{M}{C_0}, & \text{otherwise.} \end{cases} \tag{5}$$

Based on the request probability of each FG, the delay for $\mathcal{U}_j$ to download a file from $\mathcal{F}$ can be written as $D_j=\sum_{n\in\mathcal{N}}P_{\mathcal{F}_n}D_{j,n}$. Thus, the average delay for each MU can be calculated as

$$D = \frac{1}{J}\sum_{j\in\mathcal{J}} D_j. \tag{6}$$

By setting D as the OF, let us hence formulate the delay optimization problem as follows:

$$\begin{aligned} \text{minimize} \quad & D \\ \text{s.t.} \quad & \sum_{n\in\mathcal{N}}\lambda_{k,n}=1, \quad \forall\, k\in\mathcal{K}, \\ & \mathbf{\Lambda}\in\{0,1\}^{K\times N}. \end{aligned} \tag{7}$$

The optimization problem in (7) is an integer programming problem, which is NP-complete. In [14], [23], similar optimization problems have been solved by sub-optimal solutions, such as the classic greedy algorithm (GA). However, the existing solutions are typically based on centralized optimization. As we can see from (6), a centralized minimization of D at $\mathcal{B}_0$ requires the global CSI between $\mathcal{B}$ and $\mathcal{U}$, which is impractical. Hence, we will dispense with this assumption and optimize $\mathbf{\Lambda}$ in a distributed manner at a low complexity.

IV. Distributed Belief Propagation Algorithm

In this section, we propose a distributed algorithm based on BP for solving the optimization problem of (7) as follows: 1) We first develop a factor graph for describing the message passing in the BP algorithm. 2) Then we map the resultant factor graph to the network for the sake of facilitating the distributed BP optimization. 3) This solved by solving our optimization problem by proposing a distributed BP algorithm. 4) Finally, the proof of existence for a fixed point of convergence in the BP algorithm is presented.

A. Factor Graph Model

In our BP algorithm, the factor graph has to be first established based on the underlying network as a standard bipartite graphical representation of a mathematical relationship between the local delay functions and file allocation variables. Then the BP algorithm is implemented by iteratively passing messages between the local functions and their related variables. Our optimization problem is thus solved by the proposed BP algorithm based on the factor graph.

Based on the topology of the HCN, we develop a factor graph model $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ is the vertex set, and $\mathcal{E}$ is the edge set. The vertex set $\mathcal{V}$ consists of factor nodes and variable nodes. Each factor node is related to an MU and each variable node is related to an SBS. To simplify the notations, we denote by $j\in\mathcal{J}$ the j-th factor node and denote by $k\in\mathcal{K}$ the k-th variable node. Hence, the vertex set $\mathcal{V}$ is composed of $\mathcal{J}$ and $\mathcal{K}$, i.e., $\mathcal{V}=\{\mathcal{J},\mathcal{K}\}$.

As mentioned in the previous section, $\mathcal{B}_k$ will be a candidate for $\mathcal{U}_j$ to potentially associate with, but only if the received SINR at $\mathcal{U}_j$ from $\mathcal{B}_k$ is no less than the threshold δ. Correspondingly, in our factor graph, an edge in the edge set $\mathcal{E}$ connecting $\mathcal{U}_j$ and $\mathcal{B}_k$, denoted by (j,k), exists if the received SINR at $\mathcal{U}_j$ from $\mathcal{B}_k$ is no less than δ. The node k is named as a neighboring node of j, if there is an edge (j,k). Actually,

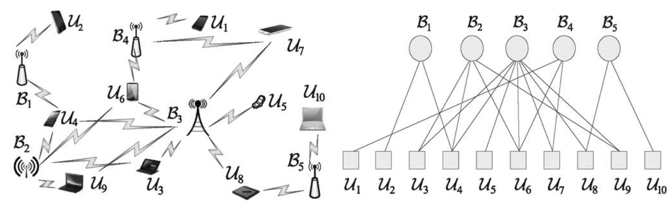


Fig. 1. Factor graph extracted from an HCN composed of 5 SBSs and 10 MUs. The edge between an SBS and an MU means that the SBS can provide a sufficiently high SINR for the MU. For instance, $\mathcal{B}_1$ can provide a sufficiently high SINR for $\mathcal{U}_2$ as well as $\mathcal{U}_4$. At the same time, $\mathcal{U}_3$ can receive a sufficiently high SINR from both $\mathcal{B}_2$ and $\mathcal{B}_3$.

$\boldsymbol{\mathcal{H}}(j)$ defined previously represents the set of the neighboring nodes of the factor node j. Furthermore, denote by $\boldsymbol{\mathcal{H}}(k)$ the set of neighboring node for the variable node k. Fig. 1 illustrates a factor graph extracted from an HCN with 5 SBSs and 10 MUs. Take $\mathcal{B}_1$ in the factor graph for example. The edges exist between $\mathcal{B}_1$ and $\mathcal{U}_2$ as well as $\mathcal{U}_4$, which means that $\mathcal{B}_1$ can provide a sufficient large SINR for both $\mathcal{U}_2$ and $\mathcal{U}_4$.

The distributed BP algorithm is based on the factor graph $\boldsymbol{\mathcal{G}}$. The factor nodes in $\boldsymbol{\mathcal{J}}$ represent the local utility functions generated from the decomposition results of the global utility function, which will be discussed later in this subsection. The variable nodes in $\boldsymbol{\mathcal{K}}$ represent the variables to be optimized, i.e., the entries of $\boldsymbol{\Lambda}$. The factor nodes and variable nodes are connected by edges in $\boldsymbol{\mathcal{E}}$, indicating the message flows in the BP algorithm. That is, messages are only passing between a node and its neighbors. We now illustrate the optimization problem on the factor graph.

1) Factor Nodes: According to Eq. (7), the OF can be decomposed into J local contributions as $D_1, \cdots, D_J$. These local contributions are calculated based on Eq. (5). Since the BP algorithm solves maximization problems, we define a series of utility functions as $F \triangleq -D$ and $F_j \triangleq -D_j$. Then our optimization problem can be rewritten as

$$\max_{\boldsymbol{\Lambda}} F(\boldsymbol{\Lambda}), \quad F = \frac{1}{J} \sum_{j \in \mathcal{J}} F_j. \tag{8}$$

We use the j-th factor node to represent the j-th local utility function F_j, which is related to $\mathcal{U}_j$. Hence, the maximization of F can be achieved by maximizing F_j at $\mathcal{U}_j, \forall j \in \boldsymbol{\mathcal{J}}$.

2) Variable Nodes: Each variable node is related to an SBS. Here, we use the k-th variable node to represent the k-th row of $\boldsymbol{\Lambda}$, denoted by $\boldsymbol{\lambda}_k$, which is related to $\mathcal{B}_k$. The location of '1' in $\boldsymbol{\lambda}_k$ indicates which specific FG is stored by $\mathcal{B}_k$. Note that the first constraint in (7) means that each SBS only stores a single FG. Given this constraint, $\boldsymbol{\lambda}_k$ has N possible values according to N different locations of '1'. We denote by $\boldsymbol{\lambda}_k^{[1]}, \cdots, \boldsymbol{\lambda}_k^{[N]}$ the N values of $\boldsymbol{\lambda}_k$. When we have $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]}$, this implies that the FG $\mathcal{F}_n$ is stored by $\mathcal{B}_k$. Take $N = 2$ for example, where $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[1]} = [1\ 0]$ indicates that the FG $\mathcal{F}_1$ is stored in the SBS $\mathcal{B}_k$, while $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[2]} = [0\ 1]$ indicates that $\mathcal{F}_2$ is stored in $\mathcal{B}_k$. The variables $\boldsymbol{\lambda}_k$, $k = 1, \cdots, K$, are the parameters to be optimized for maximizing F in (8). For simplicity, we use the matrix $\boldsymbol{\Lambda}$ to represent the set of the variables $\boldsymbol{\lambda}_k$ in the factor graph.

B. Distributed Belief Propagation

In standard BP, the variables are optimized by estimating their marginal probability distributions [32]. Note that the utility function F is a function of the file placement matrix $\boldsymbol{\Lambda}$. We define the probability mass function (PMF) $p(\boldsymbol{\Lambda})$ of $\boldsymbol{\Lambda}$ based on the utility function $F(\boldsymbol{\Lambda})$ as

$$p(\boldsymbol{\Lambda}) \triangleq \frac{1}{Z} \exp\left(\mu F(\boldsymbol{\Lambda})\right), \tag{9}$$

where μ is a positive number and Z is the normalization factor. According to [32], the result of large deviations shows that when $\mu \to \infty$, $p(\boldsymbol{\Lambda})$ concentrates around the maxima of $F(\boldsymbol{\Lambda})$, i.e., $\lim_{\mu \to \infty} \mathbb{E}(\boldsymbol{\Lambda}) = \arg\max_{\boldsymbol{\Lambda}} F(\boldsymbol{\Lambda})$, where $\mathbb{E}(\boldsymbol{\Lambda})$ is the expectation of $\boldsymbol{\Lambda}$. Once we obtain $\mathbb{E}(\boldsymbol{\Lambda})$, we can have a good estimate of the specific $\boldsymbol{\Lambda}$ which maximizes $F(\boldsymbol{\Lambda})$.

In our distributed BP, the maximization of F can be decomposed into J maximization operations on F_j at $\mathcal{U}_j, j = 1, \cdots, J$. Correspondingly, the estimation of $\boldsymbol{\Lambda}$ is decomposed into J estimations of its subsets $\boldsymbol{\Lambda}_j$ at $\mathcal{U}_j$, where $\boldsymbol{\Lambda}_j = \{\boldsymbol{\lambda}_h, \forall h \in \boldsymbol{\mathcal{H}}(j)\}$. The PMF of $\boldsymbol{\Lambda}_j$ is written as $p_j(\boldsymbol{\Lambda}_j) = \frac{1}{Z_j} \exp(\mu F_j(\boldsymbol{\Lambda}_j))$, where Z_j is the normalization factor. Since all the variables are independent, the estimation of $\boldsymbol{\Lambda}_j$ at $\mathcal{U}_j$ can be further decomposed into the estimation of each individual $\boldsymbol{\lambda}_h$ via calculating its PMF $p_j(\boldsymbol{\lambda}_h)$, which is the marginal PMF of $p_j(\boldsymbol{\Lambda}_j)$ with respect to the variable $\boldsymbol{\lambda}_h$. Hence we have $p_j(\boldsymbol{\lambda}_h) = \mathbb{E}_{\sim \boldsymbol{\lambda}_h}(p_j(\boldsymbol{\Lambda}_j))$, where $\mathbb{E}_{\sim \boldsymbol{\lambda}_h}(\cdot)$ represents the expectation over the elements in $\boldsymbol{\Lambda}_j$, except for $\boldsymbol{\lambda}_h$. The PMF $p_j(\boldsymbol{\lambda}_h)$ is viewed as the message, which is iteratively updated between $\mathcal{U}_j$ and $\mathcal{B}_h, \forall h \in \boldsymbol{\mathcal{H}}(j)$. The PMF $p_j(\boldsymbol{\lambda}_h)$ consists of N probabilities estimated by $\mathcal{U}_j$, i.e., $\Pr(\boldsymbol{\lambda}_h = \boldsymbol{\lambda}_h^{[1]}), \cdots, \Pr(\boldsymbol{\lambda}_h = \boldsymbol{\lambda}_h^{[N]})$, where $\Pr(\boldsymbol{\lambda}_h = \boldsymbol{\lambda}_h^{[n]})$ represents the probability that $\mathcal{F}_n$ is stored by $\mathcal{B}_h$.

Without a loss of generality, we assume that the edge (j, k) does exist in the factor graph. We represent the iteration index by t and denote by $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k)$ and $p_{j\to k}^{(t)}(\boldsymbol{\lambda}_k)$ the belief messages emanated from $\mathcal{B}_k$ to $\mathcal{U}_j$ and from $\mathcal{U}_j$ to $\mathcal{B}_k$ during the t-th iteration, respectively. The steps describing the distributed BP are as follows.

1) Initialization: At the variable nodes, set $t = 1$ and let $p_{k\to j}^{(1)}(\boldsymbol{\lambda}_k)$ to be the initial distribution of $\boldsymbol{\lambda}_k$, e.g., the *a priori* popularity distribution $\boldsymbol{\mathcal{P}}$.

2) Variable Node Update: During the t-th iteration, each SBS $\mathcal{B}_k$ updates the message $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k)$ to be sent to $\mathcal{U}_j$ based on the messages gleaned from $\mathcal{B}_k$'s neighboring MUs other than $\mathcal{U}_j$ in the previous iteration. This includes the calculations of N probabilities. Given $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]}, \forall n \in \boldsymbol{\mathcal{N}}$, we have

$$p_{k\to j}^{(t)}\left(\boldsymbol{\lambda}_k^{[n]}\right) = \frac{1}{Z_k} \prod_{\hbar \in \boldsymbol{\mathcal{H}}(k)\setminus\{j\}} p_{\hbar\to k}^{(t-1)}\left(\boldsymbol{\lambda}_k^{[n]}\right), \tag{10}$$

where Z_k is the normalization factor so that we have $\sum_{n\in\boldsymbol{\mathcal{N}}} p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]}) = 1$.

3) Factor Node Update: In the t-th iteration, $\mathcal{U}_j$ updates the N probabilities of the message $p_{j\to k}^{(t)}(\boldsymbol{\lambda}_k)$ to be sent to $\mathcal{B}_k$, which is based on the messages received from $\mathcal{U}_j$'s neighboring SBSs, except for $\mathcal{B}_k$. The messages updated at the factor nodes are calculated according to the marginal PMF. Given $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]}$, $\forall n \in \boldsymbol{\mathcal{N}}$, we have

$$\begin{aligned} p_{j\to k}^{(t)}\left(\boldsymbol{\lambda}_k^{[n]}\right) &= \mathbb{E}_{\sim\boldsymbol{\lambda}_k}\left(\exp\left(\mu F_j\left(\boldsymbol{\lambda}_k^{[n]}, \{\boldsymbol{\lambda}_h,\ \forall h \in \boldsymbol{\mathcal{H}}(j) \setminus \{k\}\}\right)\right)\right) \\ &= \sum_{h\in\boldsymbol{\mathcal{H}}(j)\setminus\{k\}} \sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}} \left(\prod_{q\in\boldsymbol{\mathcal{H}}(j)\setminus\{k\}} p_{q\to j}^{(t)}(\boldsymbol{\lambda}_q)\cdot \right. \\ &\qquad \left. \exp\left(\mu F_j\left(\boldsymbol{\lambda}_k^{[n]}, \{\boldsymbol{\lambda}_h,\ \forall h \in \boldsymbol{\mathcal{H}}(j) \setminus \{k\}\}\right)\right)\right). \end{aligned} \tag{11}$$

4) Final Solution: Let us assume that there are $t = T$ iterations in the distributed BP algorithm. After T iterations, the probability that $\mathcal{F}_n$ is stored by $\mathcal{B}_k$ can be obtained by

$$\Pr\left(\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]}\right) = \frac{1}{Z_k} \prod_{\hbar\in\boldsymbol{\mathcal{H}}(k)} p_{\hbar\to k}^{(T)}\left(\boldsymbol{\lambda}_k^{[n]}\right). \tag{12}$$

Based on (12), the decision as to which file should be stored by $\mathcal{B}_k$ can be made by choosing the specific file that has the maximum *a posteriori* probability $\Pr(\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{[n]})$, $\forall n \in \boldsymbol{\mathcal{N}}$.

C. Convergence to a Fixed Point

Let us now investigate the existence of a fixed point of convergence in our distributed BP algorithm. The essence of the distributed BP algorithm is to keep updating the PMF $p_j(\boldsymbol{\lambda}_k)$ before reaching its final estimate. Based on (10) and (11), the evolution of $p_j(\boldsymbol{\lambda}_k)$ during the t-th iteration can be obtained from the PMFs in the $(t-1)$-th iteration as

$$\begin{aligned} p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k) = \frac{1}{Z_k} \prod_{\hbar\in\boldsymbol{\mathcal{H}}(k)\setminus\{j\}} \sum_{h\in\boldsymbol{\mathcal{H}}(\hbar)\setminus\{k\}} \sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}} \\ \left(\exp\left(\mu F_{\hbar}(\boldsymbol{\Lambda}_{\hbar})\right)\cdot \prod_{q\in\boldsymbol{\mathcal{H}}(\hbar)\setminus\{k\}} p_{q\to\hbar}^{(t-1)}(\boldsymbol{\lambda}_q)\right). \end{aligned} \tag{13}$$

We view the PMF $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k)$ as a probability vector of length N. We define the probability vector set $\boldsymbol{\mathcal{M}}^{(t)} \triangleq \left\{p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k)\right\}$ for all $k \in \boldsymbol{\mathcal{K}}$ as well as $j \in \boldsymbol{\mathcal{J}}$, and define the message mapping function $\boldsymbol{\Gamma} : \mathbb{R}^{N\times KJ} \to \mathbb{R}^{N\times KJ}$ based on (13) so that $\boldsymbol{\mathcal{M}}^{(t)} = \boldsymbol{\Gamma}(\boldsymbol{\mathcal{M}}^{(t-1)})$. Then we have the following lemma.

Lemma 1: The message mapping function $\boldsymbol{\Gamma}$ is a continuous mapping.

Proof: Please refer to Appendix A.

Given Lemma 1, we have the following theorem.

Theorem 1: A fixed point of convergence exists for the proposed distributed BP algorithm.

Proof: Please refer to Appendix B.

The question of convergence to the fixed point is, unfortunately, not well understood in general [24]. Generally, if the factor graph contains no cycles, the belief propagation can be shown to converge to a fixed solution point in a finite number of iterations. The performance, including the optimality and the convergence rate, of the BP crucially depends on the choice of the objective function, as well as the scale, the sparsity and the number of cycles in the underlying factor graph. As such, the theoretical analysis of the BP algorithm's optimality and convergence rate remains an open challenge.

V. A Heuristic BP With Reduced Complexity

In the context of the BP algorithm, the message $p_j(\boldsymbol{\lambda}_k)$ exchanged between $\mathcal{U}_j$ and $\mathcal{B}_k$ in each iteration, includes N probability values, which are real numbers. Hence, the communication overhead of the message passing is relatively high. Hence, we propose a heuristic BP (HBP) algorithm for reducing the communication overhead imposed. The rationale behind the term "heuristic BP" is that we still follow the classic concept of belief propagation, but use a different format of the beliefs from the conventional one.

Assuming that the edge (j, k) exists, in the t-th iteration of the HBP, instead of forwarding the N probabilities stored in $p_{j\to k}^{(t)}(\boldsymbol{\lambda}_k)$ to $\mathcal{B}_k$, $\mathcal{U}_j$ randomly selects an FG according to these N probabilities. Then the integer index $n_{j\to k}^{(t)}$ of the FG selected will be forwarded to the SBS $\mathcal{B}_k$.

At the SBS side, the SBS $\mathcal{B}_k$ receives $|\boldsymbol{\mathcal{H}}(k)|$ integers, i.e., $n_{\hbar\to k}^{(t)}$, $\forall \hbar \in \boldsymbol{\mathcal{H}}(k)$, from its neighboring MUs, where $|\cdot|$ denotes the cardinality of a set. Based on $n_{\hbar\to k}^{(t)}$, the SBS $\mathcal{B}_k$ infers the number of those MUs, which indicate that $\mathcal{F}_n$ should be stored in the SBS $\mathcal{B}_k$, for $n = 1, \cdots, N$. Let us assume now that in the t-th iteration, there are $J_{k,n}^{(t)}$ MUs specifically indicating that $\mathcal{F}_n$ should be stored in $\mathcal{B}_k$, where we have $\sum_{n\in\boldsymbol{\mathcal{N}}} J_{k,n}^{(t)} = |\boldsymbol{\mathcal{H}}(k)|$. We can view $\frac{J_{k,n}^{(t)}}{|\boldsymbol{\mathcal{H}}(k)|}$ as the probability that the specific FG $\mathcal{F}_n$ is stored by the SBS $\mathcal{B}_k$.

In this case, the probability $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ in (10) will be recalculated as

$$p_{k\to j}^{(t)}\left(\boldsymbol{\lambda}_k^{[n]}\right) = \begin{cases} \frac{J_{k,n}^{(t-1)}-1}{|\boldsymbol{\mathcal{H}}(k)|-1}, & \text{if } n = n_{j\to k}^{(t-1)}, \\ \frac{J_{k,n}^{(t-1)}}{|\boldsymbol{\mathcal{H}}(k)|-1}, & \text{if } n \neq n_{j\to k}^{(t-1)}. \end{cases} \tag{14}$$

Note that in (14), the information $n_{j\to k}^{(t-1)}$ transmitted from the MU $\mathcal{U}_j$ to the SBS $\mathcal{B}_k$ is excluded when calculating $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$, for the sake of ensuring that only uncorrelated information is exchanged throughout the HBP.

At the MU side, it is clear that the MU $\mathcal{U}_j$ has to obtain $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ for the sake of updating the output information. However, there is no need for the SBS $\mathcal{B}_k$ to transmit the N probabilities $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ to each of its neighboring MUs. Alternatively, $\mathcal{B}_k$ broadcasts the N integers, $J_{k,1}^{(t)}, \cdots, J_{k,N}^{(t)}$ to the neighboring MUs for reducing the transmission overhead. After receiving the N integers from the SBS $\mathcal{B}_k$, the MU $\mathcal{U}_j$ calculates $p_{k\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ in (14).

Based on the above discussions, the HBP algorithm can be summarized as follows.

1) Initialization: At the variable nodes, we set $t = 1$. The SBS $\mathcal{B}_k$ randomly generates $|\boldsymbol{\mathcal{H}}(k)|$ independent integers, $n_1, \cdots, n_{|\boldsymbol{\mathcal{H}}(k)|}$, according to the popularity distribution $\boldsymbol{\mathcal{P}}$. These integers are viewed as the indexes of the FGs. We then set $J_{n,k}^{(1)}$ to be the number of the integers that are equal to n.

2) Variable Node Update: In the t-th iteration, $\mathcal{B}_k$ updates and broadcasts the N integers $J_{n,k}^{(t)}$, for $n = 1, \cdots, N$, to the neighboring MUs. The resulting calculations performed on these N integers $J_{n,k}^{(t)}$ are based on the integers $n_{\hbar\to k}^{(t-1)}$, $\forall \hbar \in \boldsymbol{\mathcal{H}}(k)$, received from the neighboring MUs during the last iteration. Specifically, the n-th integer $J_{n,k}^{(t)}$ is obtained by counting the number of $n_{\hbar\to k}^{(t-1)}$ that are equal to n.

3) Factor Node Update: The MU $\mathcal{U}_j$ first calculates the probabilities $p_{h\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$, $\forall h \in \boldsymbol{\mathcal{H}}(j)$ according to Eq. (14) based on the integers gleaned from the SBS $\mathcal{B}_h$. Then based on $p_{h\to j}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$, $\forall h \in \boldsymbol{\mathcal{H}}(j) \setminus \{k\}$, $\mathcal{U}_j$ calculates $p_{j\to k}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$ according to Eq. (11). After obtaining the N probabilities $p_{j\to k}^{(t)}(\boldsymbol{\lambda}_k^{[n]})$, $n = 1, \cdots, N$, $\mathcal{U}_j$ randomly chooses an FG according to these N probabilities and sends the index $n_{j\to k}^{(t)}$ of the FG to the SBS $\mathcal{B}_k$.

4) Final Solution: After T iterations, the SBS $\mathcal{B}_k$ makes the decision that the FG $\mathcal{F}_{\hat{n}}$ should be stored for ensuring that

$$\hat{n} = \arg\max_{n\in\mathcal{N}} J_{k,n}^{(T)}. \tag{15}$$

The overhead of the HBP is significantly lower than that of the original BP introduced in the previous section. From a communication complexity perspective, in each iteration of the HBP, an SBS $\mathcal{B}_k$ broadcasts N integers, while an MU $\mathcal{U}_j$ transmits $|\boldsymbol{\mathcal{H}}(j)|$ integers. On the other hand, in the original BP, $\mathcal{B}_k$ transmits $N|\boldsymbol{\mathcal{H}}(k)|$ real numbers, while $\mathcal{U}_j$ transmits $N|\boldsymbol{\mathcal{H}}(j)|$ real numbers for each iteration. From a computational complexity perspective, in a single iteration of the HBP, the computational complexity is on the order of $O(N)$ at the SBS $\mathcal{B}_k$, and $O(|\mathcal{H}(j)|N^{|\mathcal{H}(j)|})$ at the MU $\mathcal{U}_j$. On the other hand, in the original BP, the computational complexity is $O(N|\mathcal{H}(k)|^2)$ at $\mathcal{B}_k$, and $O(|\mathcal{H}(j)|N^{|\mathcal{H}(j)|})$ at $\mathcal{U}_j$ for each iteration.

VI. Performance Analysis Based on Stochastic Geometry

In this section, we analyze both the average degree distribution of the factor graph and the average downloading performance based on stochastic geometry theory. We model the distribution of the MUs as a PPP Φ_U having the intensity of λ_U, and that of the SBSs as an independent PPP Φ_B with the intensity λ_B [31], [33]. For simplicity, we assume that all the SBSs have the same transmission power P. In the following, both the degree distribution and the downloading performance are averaged over both the channels' fading coefficients and over the PPP distributions of the nodes.

A. Average Degree Distributions of the Factor Graph

Let us now investigate the degree distribution of the factor graph averaged over PPP. Note that the degree of a factor node j is defined as the number of its neighboring variable nodes, given by the cardinality $|\boldsymbol{\mathcal{H}}(j)|$, while the degree of a variable node k is defined as the number of its neighboring factor nodes, i.e., $|\boldsymbol{\mathcal{H}}(k)|$. Then we have the following theorem.

Theorem 2: The factor nodes in the factor graph have the average degree

$$\zeta_U = 2\pi\lambda_B Z(\lambda_B, P, \alpha, \delta), \tag{16}$$

and the variable nodes have the average degree

$$\zeta_B = 2\pi\lambda_U Z(\lambda_B, P, \alpha, \delta), \tag{17}$$

where we have

$$Z(\lambda_B, P, \alpha, \delta) = \int_0^\infty \exp\left\{-\frac{2\lambda_B\pi}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) r^2 - \frac{\delta\sigma^2}{P} r^\alpha\right\} r\mathrm{d}r \tag{18}$$

and the Beta function $B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}\mathrm{d}t$.

Proof: Please refer to Appendix C.

When neglecting the noise, we have the following corollary based on *Theorem 2*.

Corollary 1: When neglecting the noise, $Z(\lambda_B, P, \alpha, \delta)$ in (18) can be rewritten as

$$Z(\lambda_B, P, \alpha, \delta) = \frac{\alpha}{4\pi\lambda_B B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)\delta^{\frac{2}{\alpha}}}. \tag{19}$$

Then we can simplify the average degree of the factor nodes in Eq. (16) to

$$\zeta_U = \frac{\alpha}{2\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)}, \tag{20}$$

and the average degree of the variable nodes in Eq. (17) to

$$\zeta_B = \frac{\lambda_U\alpha}{2\lambda_B\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)}. \tag{21}$$

Proof: Please refer to Appendix D.

Equations (20) and (21) can be seen as approximations of (16) and (17), respectively, when the effects of the noise are neglected. These approximations are significantly accurate for the HCN, since the interference effects are dominant due to the dense deployments of the SBSs.

From (20), we can see that ζ_U is only related to δ and α, but is independent of λ_U, P and λ_B. In other words, the factor node degree has no relation with the intensities of the MUs and SBSs or with the power of the SBSs. The intuitive reason is that although increasing both the PPP intensities and the power of the SBSs can increase the total signal power, the interference also increases at the same time, which keeps the degree ζ_U of the factor nodes constant. Similarly, observe from (21) that ζ_B is independent of the power P, i.e., increasing the transmission power of the SBSs will not influence the average degree distribution of the factor graph.

Remark 1: We observe that $B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) = \pi$ when $\alpha = 4$. Thus, we have closed-form expressions for ζ_U and ζ_B in (20) and (21), respectively, when $\alpha = 4$.

B. Downloading Performance of Random Caching

Since the performance of BP based caching remains difficult for mathematical analysis in closed form, we propose a random caching scheme and analyze its performance based on stochastic geometry theory. The random caching is realized by randomly picking out $\Omega_{\mathcal{F}_n} \cdot K$ $(0 \le \Omega_{\mathcal{F}_n} \le 1)$ SBSs from the entire set of K SBSs for caching the FG $\mathcal{F}_n$.

To evaluate the downloading performance, we first define an outage $\mathcal{Q}_n$ as the event of an MU's failing to find the FG $\mathcal{F}_n$ in its neighboring SBSs. The following theorem states an upper bound of the OP of $\mathcal{Q}_n$. As mentioned before, since the interference is the dominant factor predetermining the network performance, we ignore the noise effects in the following performance analysis to simplify our derivations.

Theorem 3: The OP for downloading a file in $\mathcal{F}_n$ can be upper-bounded by

$$\Pr(\mathcal{Q}_n) \le \frac{C(\delta,\alpha)\left(1-\Omega_{\mathcal{F}_n}\right) + A(\delta,\alpha)\Omega_{\mathcal{F}_n}}{C(\delta,\alpha)\left(1-\Omega_{\mathcal{F}_n}\right) + A(\delta,\alpha)\Omega_{\mathcal{F}_n} + \Omega_{\mathcal{F}_n}}, \quad (22)$$

where we have $C(\delta,\alpha) \triangleq \frac{2}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)$, $A(\delta,\alpha) \triangleq \frac{2\delta}{\alpha-2} {}_2F_1\left(1, 1-\frac{2}{\alpha}; 2-\frac{2}{\alpha}; -\delta\right)$, and ${}_2F_1$ represents the hypergeometric function.

Proof: Please refer to Appendix E.

When the path-loss exponent $\alpha = 4$, we have $C(\delta, 4) = \frac{\sqrt{\delta}}{2}\pi$ and $A(\delta, 4) = \delta {}_2F_1\left(1, \frac{1}{2}; \frac{3}{2}, -\delta\right)$. It becomes clear from (22) that $\Pr(\mathcal{Q}_n)$ is only related to δ and $\Omega_{\mathcal{F}_n}$, where a higher δ leads to a higher $\Pr(\mathcal{Q}_n)$. This is because a larger δ will reduce the number of possibly eligible serving SBSs, resulting in an increase of OP. We can see that a higher $\Omega_{\mathcal{F}_n}$ leads to a lower $\Pr(\mathcal{Q}_n)$.

Let us define the averaged OP $\mathcal{Q}$ over all the files. Based on the file popularity, the OP of $\mathcal{Q}$ can be upper-bounded by

$$\begin{aligned}\Pr(\mathcal{Q}) &= \sum_{n\in\mathcal{N}} P_{\mathcal{F}_n} \Pr(\mathcal{Q}_n) \\ &\le \sum_{n\in\mathcal{N}} \frac{P_{\mathcal{F}_n}\left(C(\delta,\alpha)\left(1-\Omega_{\mathcal{F}_n}\right) + A(\delta,\alpha)\Omega_{\mathcal{F}_n}\right)}{C(\delta,\alpha)\left(1-\Omega_{\mathcal{F}_n}\right) + A(\delta,\alpha)\Omega_{\mathcal{F}_n} + \Omega_{\mathcal{F}_n}}. \quad (23)\end{aligned}$$

The average delay $\bar{D}$ of each MU can be obtained based on the average OP, i.e.,

$$\bar{D} = (1 - \Pr(\mathcal{Q}))\,\bar{D}_s + \Pr(\mathcal{Q})\frac{M}{C_0}, \quad (24)$$

where $\bar{D}_s$ is the average delay of downloading from the SBSs. The delay $\bar{D}$ can be seen as the average value of D in Eq. (6) over both the PPP and the channel fading. Note that $\bar{D}_s$ is usually challenging to calculate and does not have a closed form in the PPP analysis.

Next, we optimize $\Omega_{\mathcal{F}_n}$ for improving the downloading performance. Since we do not have a closed-form expression for $\bar{D}$, we minimize the upper bound of $\Pr(\mathcal{Q})$ in (23), i.e.,

$$\begin{aligned}&\max_{\{\Omega_{\mathcal{F}_n}\}} \sum_{n\in\mathcal{N}} \frac{P_{\mathcal{F}_n}\Omega_{\mathcal{F}_n}}{\Omega_{\mathcal{F}_n}\left(A(\delta,\alpha) - C(\delta,\alpha) + 1\right) + C(\delta,\alpha)}, \\ &\text{s.t.} \quad \sum_{n\in\mathcal{N}} \Omega_{\mathcal{F}_n} = 1, \\ &\qquad\quad \Omega_{\mathcal{F}_n} \ge 0. \end{aligned} \quad (25)$$

By relying on the classic Lagrangian multiplier, we arrive at the optimal solution as

$$\Omega^{\star}_{\mathcal{F}_n} = \max\left\{\frac{\sqrt{\frac{P_{\mathcal{F}_n}}{\xi}} - C(\delta,\alpha)}{A(\delta,\alpha) - C(\delta,\alpha) + 1}, 0\right\}, \quad (26)$$

where $\xi = \frac{\left(\sum_{q=1}^{n^*}\sqrt{P_{\mathcal{F}_q}}\right)^2}{(n^* C(\delta,\alpha_s) + A(\delta,\alpha_s) - C(\delta,\alpha_s) + 1)^2}$, and n^* satisfies the constraint that $\Omega_{\mathcal{F}_n} \ge 0$.

VII. Simulation Results

In this section, we first focus on the HCNs associated with PPP distributed nodes, where we investigate the average degree distribution of the factor graph and the performance of the random caching scheme. Then we consider an HCN supporting a fixed number of nodes. We investigate the delay optimized by the BP algorithm and compare it to other benchmarks, including both the random caching and the optimal scheme using exhaustive search.

Note that the physical layer parameters in our simulations, such as the path-loss exponent, noise power, transmit power of the SBSs, and the intensity of the SBSs, are chosen to be practical and in line with the values set by 3GPP standards. For instance, the transmit power of an SBS is typically 2 Watt in 3GPP. The unit of power, such as noise power and transmit power, is the classic Watt. The intensities of the SBSs and MUs are expressed in terms of the numbers of the nodes per square kilometer. Unless specified otherwise, we set the path loss to $\alpha = 4$, the number of files to $Q = 100$, transmit power to $P = 2$, and the noise power to $\sigma^2 = 10^{-10}$. All the simulations are executed with MATLAB. Also, we consider the performance averaged over a thousand network cases, where the locations of network nodes are uniformly distributed in each case, and randomly changed from case to case.

A. Average Degree Distributions of Factor Graph

We compare our Monte-Carlo simulations and analytical results in the HCNs at various transmission powers and node densities. Fig. 2 shows the average degree of the factor nodes with different transmission power P, SBSs' intensity λ_B, and MUs' intensity λ_U. We can see that for a given δ, the degree ζ_U remains unaffected by the specific choice of P, λ_B, and λ_U. Observe that our analytical results are consistent with the simulations. Similarly, Fig. 3 shows the average degree of

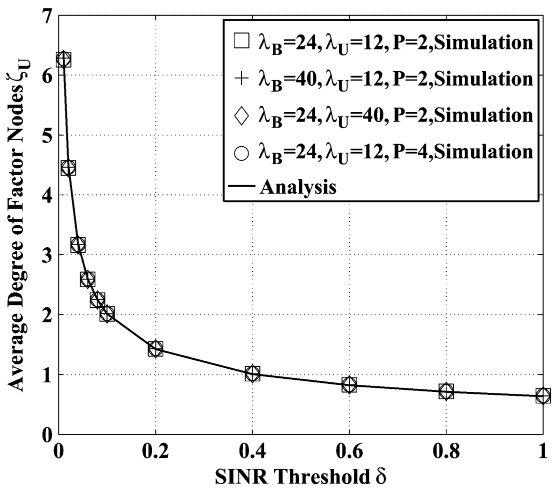


Fig. 2. Average degree of factor nodes ζ_U vs. δ for different SBS and MU intensities of λ_B and λ_U, and for transmit powers of $P = 2$ and 4.

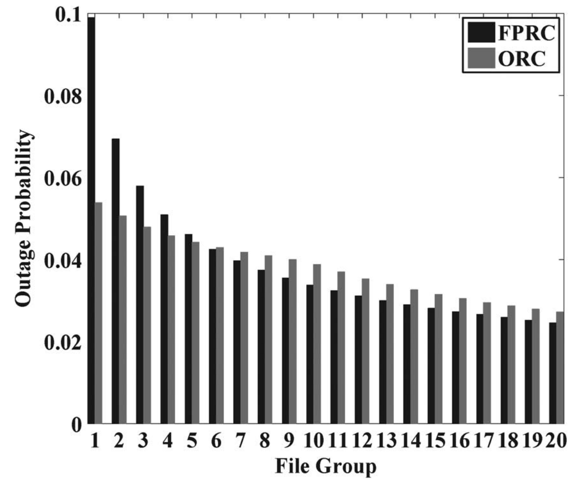


Fig. 4. Outage probabilities $\Pr(\mathcal{Q}_n) \cdot P_{\mathcal{F}_n}$ for individual FGs $\mathcal{F}_n$ under the file popularity based random caching (FPRC) and optimized random caching (ORC) schemes.

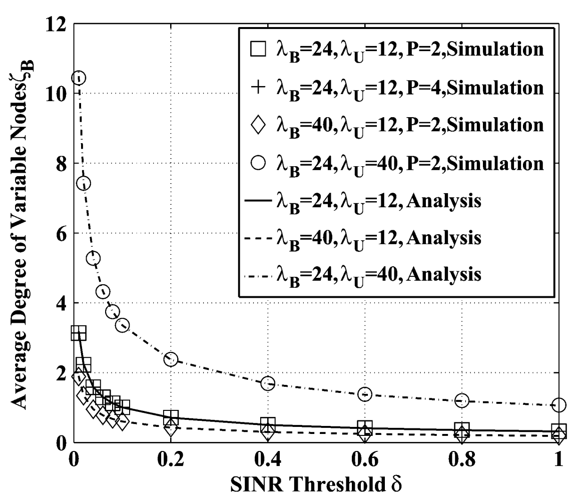


Fig. 3. Average degree of variable nodes ζ_B vs. δ for different SBS and MU intensities of λ_B and λ_U, and for transmit powers of $P = 2$ and 4.

the variable nodes of different powers and node intensities, demonstrating that the results are independent of the power P, but depend on the densities λ_B and λ_U. We can also see that the analytical results match well with the simulation results.

B. Average Downloading Performance of Random Caching

Let us now evaluate the average downloading performance of the random caching scheme supporting PPP distributed nodes. The file distribution $\boldsymbol{\mathcal{P}} = \{p_1, \cdots, p_Q\}$ is modeled by the Zipf distribution [34], which can be expressed as

$$p_f = \frac{1/f^s}{\sum_{q=1}^{Q} 1/q^s}, \quad \text{for } f = 1, \cdots, Q, \tag{27}$$

where the exponent $0 < s \leq 1$ is a real number, and it characterizes the popularity of files. Explicitly, a larger s corresponds to a higher content reuse, i.e., the most popular files account for the majority of requests. Note that $P_{\mathcal{F}_n}$ can be obtained based on p_f via Eq. (2).

For the simulation results of this subsection, we assume that each SBS caches $G = 5$ files, hence there are $N = Q/G = 20$ FGs. We commence by considering the OP. In our optimized random caching (ORC), we set $\Omega_{\mathcal{F}_n}$ as in (26). For comparison, we also consider another random caching scheme from [19] as our the benchmark, namely, the file popularity based random caching (FPRC). In the FPRC, $\Omega_{\mathcal{F}_n}$ is chosen to be consistent with the file popularity, i.e., we have $\Omega_{\mathcal{F}_n} = P_{\mathcal{F}_n}$.

Fig. 4 shows the OPs $\Pr(\mathcal{Q}_n) \cdot P_{\mathcal{F}_n}$ for individual FGs under both the ORC and the FPRC schemes, where we have $\delta = 0.03$ and $s = 0.5$. The conditional OP $\Pr(\mathcal{Q}_n)$ (given a file in $\mathcal{F}_n$ is requested) is calculated from Eq. (22), while the request probability $P_{\mathcal{F}_n}$ of $\mathcal{F}_n$ is calculated from Eq. (2). The FGs are arranged in descending order of popularity, i.e., the first FG has the highest popularity, while the last one has the lowest popularity. We can see from the figure that compared to the FPRC, FGs having a higher popularity have a lower OP, while the ones with lower popularity have higher OPs in the ORC. For example, the OP for the most popular FG is around 0.054 in the ORC in contrast to 0.099 in the FPRC, while the probability of the least popular FG is 0.27 in the ORC in contrast to 0.25 in the FPRC. This is because the ORC is reminiscent of the classic water-filling, allocating more SBSs for caching the higher popular FGs for ensuring the minimization of the average OP.

Let us now investigate the average OP $\Pr(\mathcal{Q})$. Figs. 5 and 6 show $\Pr(\mathcal{Q})$ for different δ and s values, respectively. In Fig. 5, we fix $s = 0.5$, while in Fig. 6, we fix $\delta = 0.03$. The dashed lines with different marks are based on the simulations associated with various power and densities, while the solid lines represent the analytical upper bounds of Eq. (23). We can see that the average OP is independent of both the power P and densities λ_B and λ_U. The ORC scheme has a lower average OP than the FPRC. Furthermore, as expected, a higher SINR threshold δ leads to a higher OP, as shown in Fig. 5. At the same time, it is interesting to observe from Fig. 6 that a larger s, representing more imbalanced downloading requests on the different files, can dramatically reduce the OP. We can see that the upper bounds evaluated from Eq. (23) match the simulations quite accurately.

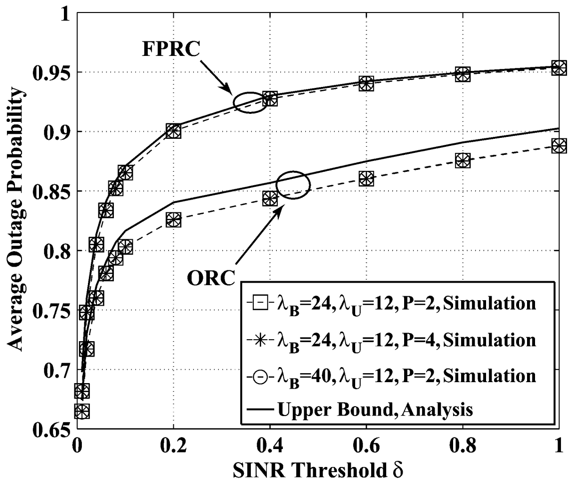


Fig. 5. Average outage probabilities $\Pr(\mathcal{Q})$ vs. δ under the FPRC and ORC schemes for different SBS and MU intensities λ_B and λ_U, and for transmit powers $P = 2$ and 4.

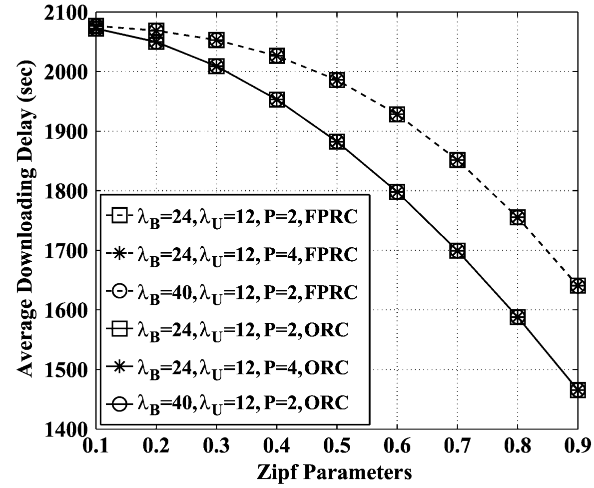


Fig. 7. Average downloading delay $\bar{D}$ vs. the Zipf parameter s under the FPRC and ORC schemes for different SBS and MU intensities λ_B and λ_U, and for transmit powers $P = 2$ and 4.

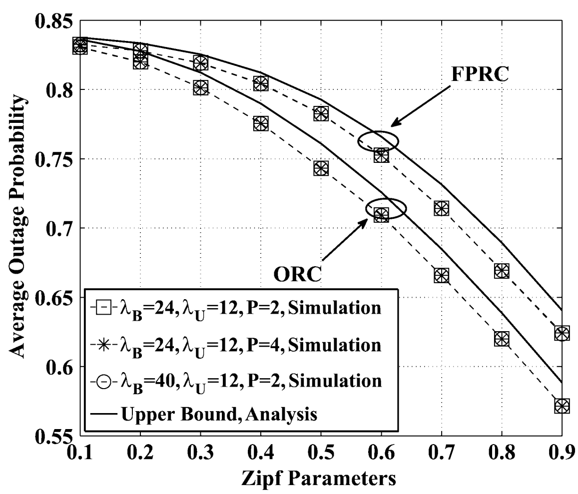


Fig. 6. Average outage probabilities $\Pr(\mathcal{Q})$ vs. the Zipf parameter s under the FPRC and ORC schemes for different SBS and MU intensities λ_B and λ_U, and for transmit powers $P = 2$ and 4.

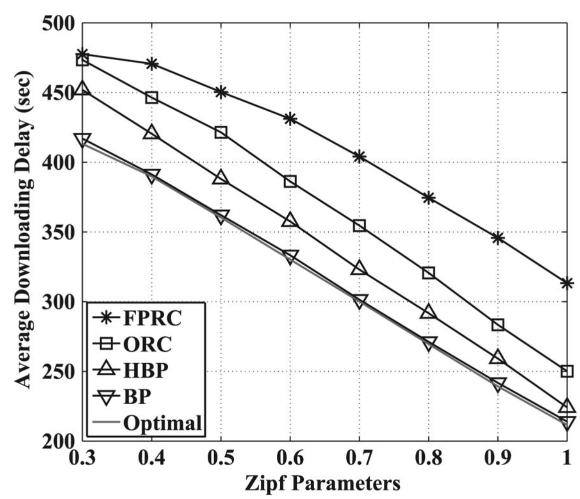


Fig. 8. Average downloading delay $\bar{D}$ vs. the Zipf parameter s under various schemes in the first scenario.

Next, we consider the average delay $\bar{D}$ in Eq. (24), where we assume an SINR threshold of $\delta = 0.03$, a bandwidth of $W = 10^7$ Hz, and a file size of $M = 10^9$ bits. Since C_0 should be always less than the maximum possible downloading rate provided by the SBSs, we assume $C_0 = W \log(1 + \delta)$. For $\delta = 0.03$, C_0 becomes 4.26×10^5 bits/sec. Fig. 7 illustrates the average downloading delay associated with different s values. We can see that the ORC scheme always outperforms the FPRC scheme, and that their performance gap becomes larger upon increasing s. Again, the observed performance does not depend on the powers and intensities of the nodes.

C. Delay Performance of Distributed BP Algorithms

Let us now study the delay performance of distributed BP-based optimizations. We consider HCNs having fixed numbers of SBSs and MUs, where the locations of these nodes are time-variant. We first consider a small network, in which the optimal solution is found with the aid of an exhaustive search. This will allow us to characterize the performance disparity between the proposed BP algorithm and the optimal search-based solution. Then we focus our attention on a larger network to show the robustness of our BP algorithms. In both scenarios, we set the SINR threshold to $\delta = 0.1$, the transmission power to $P = 2$, the bandwidth to $W = 10^7$ Hz, and the file size to $M = 10^9$ bits. Similar to the previous subsection, we assume that the rate provided by the MBS as $C_0 = W \log(1 + \delta)$. For $\delta = 0.1$, we have C_0 as 1.3×10^6 bits/sec.

In the first scenario, the nodes are arranged in a 0.6×0.6 km^2 area using 8 SBSs and 4 MUs. We assume that each SBS caches $G = 25$ files, and there are $N = Q/G = 4$ FGs. Fig. 8 shows the average delay performance under various schemes, where 'HBP' is the heuristic BP algorithm proposed in Section V, 'BP' is the original BP algorithm proposed in Section IV, and 'Optimal' is the optimal scheme relying on an exhaustive search. We can see from Fig. 8 that the original BP approaches the optimal scheme within a small delay margin. The proposed HBP performs slightly worse than the original BP, with a

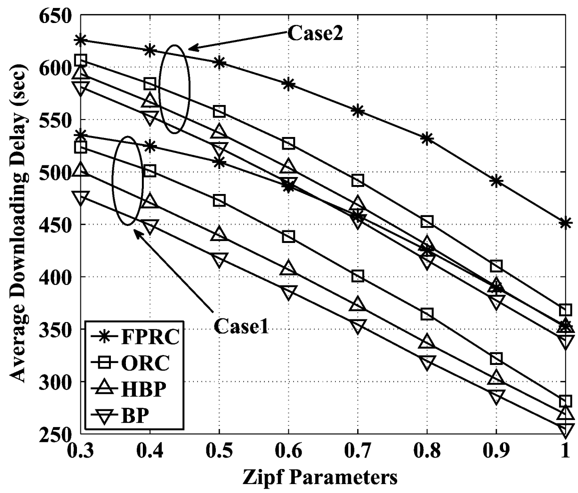


Fig. 9. Average downloading delay $\bar{D}$ vs. the Zipf parameter s under various schemes in the second scenario.

relatively modest delay degradation of around 5% or 20 seconds, while it outperforms the ORC scheme by about 10% or 40 seconds gain. The FPRC performs the worst among all the caching schemes, exhibiting a substantial delay gap between the FPRC scheme and the ORC scheme.

In the second scenario, the nodes are arranged in a 1.5×1.5 km^2 area with 50 SBSs and 25 MUs. We consider two cases, namely Case1 and Case2. In Case1, we assume that each SBS caches $G = 20$ files and there are $N = Q/G = 5$ FGs, while in Case2, we assume that each SBS caches $G = 10$ files and that we have $N = Q/G = 10$. Fig. 9 shows the average delay performance under various schemes. It is clear from Fig. 9 that in both cases the BP algorithm performs the best, while the FPRC performs the worst. The HBP exhibits a tiny delay increase of around 3% performance loss compared to the original BP, although it dramatically reduces the communication complexity during the optimization process.

Note also in Fig. 9 that the ORC suffers from a 5% performance loss compared to the HBP, but it is much less complex than the HBP and BP. The optimization in ORC is based on the statistical information available about both of channels and the locations of the nodes, while both the BP and the HBP exploit the relevant instantaneous information at a relatively high communication complexity. In this sense, the ORC constitutes an efficient caching scheme. Furthermore, we can see from Fig. 9 that there is a tradeoff between the storage and delay, i.e., a larger storage at each SBS in Case1 leads to a lower downloading delays compared to Case2.

In the above BP simulations, we set the maximum number of iterations to $T = 15$. Table I shows the average number of iterations under different s values for the two scenarios. We can see that the HBP relies on more iterations than the BP. Nevertheless, the overall communication complexity of the HBP is still lower than that of the BP, as we have discussed in Section V. Explicitly, for each iteration of the HBP, $\mathcal{B}_k$ broadcasts N integers and $\mathcal{U}_j$ transmits $|\boldsymbol{\mathcal{H}}(j)|$ integers. By contrast, in the original BP, $\mathcal{B}_k$ transmits $N|\boldsymbol{\mathcal{H}}(k)|$ real numbers and $\mathcal{U}_j$ transmits $N|\boldsymbol{\mathcal{H}}(j)|$ real numbers.

TABLE I
THE AVERAGE NUMBER OF ITERATIONS UNDER DIFFERENT s

	Zipf Parameter s							
s	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Average Number of Iterations for Scenario 1								
BP	4.466	4.406	4.002	3.652	3.574	3.412	3.12	2.862
HBP	8.431	8.235	7.634	7.094	6.71	6.494	6.097	5.263
Average Number of Iterations for Scenario 2								
Case1								
BP	9.429	8.412	7.632	7.326	6.576	5.978	5.804	5.696
HBP	14.973	14.903	14.817	14.783	14.722	14.667	14.623	14.443
Case2								
BP	9.548	8.642	7.987	7.483	7.119	6.746	6.057	5.841
HBP	14.994	14.97	14.925	14.821	14.877	14.722	14.648	14.549

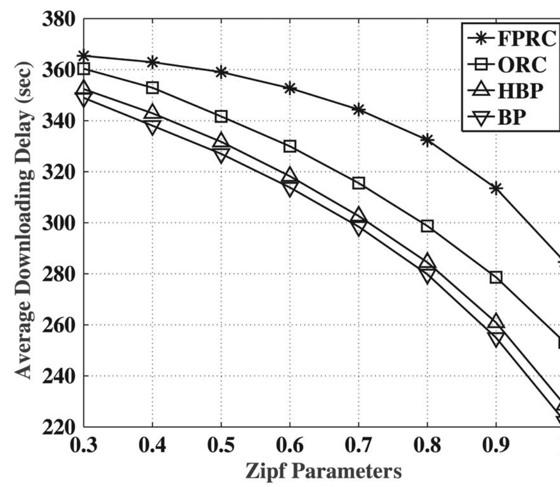


Fig. 10. Average downloading delay $\bar{D}$ vs. the Zipf parameter s under various schemes in the large scale network.

D. Delay Performance in a Large Scale Network

Finally, we consider a large-scale network associated with $Q = 1000$ files, 50 SBSs, and 100 MUs within an area of 5×5 km^2. Furthermore, we consider a lower connection probability to the SBSs by setting $\delta = 0.2$. By assuming that each SBS is capable of caching 20 files, we have overall 50 file groups. Fig. 10 shows the average delay performance. We can see from the figure that both BP algorithms perform better than the random caching schemes. Particularly, the HBP has a roughly 1% performance loss compared to the original BP, which imposes however a much reduced communication complexity. This implies that our BP algorithms are robust in large-scale networks associated with a large number of files and network nodes.

Further comparing Figs. 8, 9, and 10, it is interesting to observe that the gap between our BP and HBP algorithms becomes smaller when the network scale becomes larger. More particularly in Fig. 10, the performance of these two schemes almost overlaps. This indicate that in large scale networks, we may consider to use the HBP rather than BP to obtain a good performance at a much reduced complexity.

VIII. CONCLUSION

In this paper, we designed distributed caching optimization algorithms with the aid of BP for minimizing the downloading latency in HCNs. Specifically, a distributed BP algorithm was proposed based on the factor graph according to the network structure. We demonstrated that a fixed point of convergence exists for the distributed BP algorithm. Furthermore, we proposed a modified heuristic BP algorithm for further reducing the complexity. To have a better understanding of the average network performance under varying numbers and locations of the network nodes, we involved stochastic geometry theory in our performance analysis. Specifically, we developed the average degree distribution of the factor graph, as well as an upper bound of the OP for random caching schemes. The performance of the random caching was also optimized based on the upper bound derived. Simulations showed that the proposed distributed BP algorithm approaches the optimal performance of the exhaustive search within a small margin, while the modified BP offers a good performance at a very low complexity. Additionally, the average performance obtained by stochastic geometry analysis matches well with our Monte-Carlo simulations, and the optimization based on the upper bound derived provides a better performance than the benchmark of [19].

Appendix A
Proof of Lemma 1

To simplify the notation in the proof, we assume that $\mathcal{H}(j) = \mathcal{K}, \forall j \in \mathcal{J}$ and $\mathcal{H}(k) = \mathcal{J}, \forall k \in \mathcal{K}$. Consider a pair of probability vector sets $\mathcal{M}^{(t-1)} = \left\{p_{k\to j}^{(t-1)}(\boldsymbol{\lambda}_k)\right\}$ and $\widetilde{\mathcal{M}}^{(t-1)} = \left\{\tilde{p}_{k\to j}^{(t-1)}(\boldsymbol{\lambda}_k)\right\}$. Then we have the supremum norm

$$\begin{aligned}
&\left\|\boldsymbol{\Gamma}\left(\mathcal{M}^{(t-1)}\right) - \boldsymbol{\Gamma}\left(\widetilde{\mathcal{M}}^{(t-1)}\right)\right\|_{\sup}\\
&= \max_{k,j,n}\left|p_{k\to j}^{(t)}\left(\boldsymbol{\lambda}_k^{[n]}\right) - \tilde{p}_{k\to j}^{(t)}\left(\boldsymbol{\lambda}_k^{[n]}\right)\right|\\
&= \max_{k,j,n}\left|\prod_{i\in\mathcal{J}\setminus\{j\}}\sum_{h\in\mathcal{K}\setminus\{k\}}\sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}}\left(\exp\left(\mu F_i(\boldsymbol{\Lambda}_i)\right)\left(\prod_{q\in\mathcal{K}\setminus\{k\}}\right.\right.\right.\\
&\qquad\left.\left.\left. p_{q\to i}^{(t-1)}(\boldsymbol{\lambda}_q) - \prod_{q\in\mathcal{K}\setminus\{k\}}\tilde{p}_{q\to i}^{(t-1)}(\boldsymbol{\lambda}_q)\right)\right)\right|\\
&\overset{(a)}{\le} \max_j \prod_{i\in\mathcal{J}\setminus\{j\}}\sum_{h\in\mathcal{K}\setminus\{k\}}\sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}}\\
&\qquad\left|\prod_{q\in\mathcal{K}\setminus\{k\}} p_{q\to i}^{(t-1)}(\boldsymbol{\lambda}_q) - \prod_{q\in\mathcal{K}\setminus\{k\}}\tilde{p}_{q\to i}^{(t-1)}(\boldsymbol{\lambda}_q)\right|\\
&\overset{(b)}{\le} (K-1)N^{K-1}\max_j\\
&\qquad\prod_{i\in\mathcal{J}\setminus\{j\}}\max_{q\in\mathcal{K}\setminus\{k\},n}\left|p_{q\to i}^{(t-1)}\left(\boldsymbol{\lambda}_q^{[n]}\right) - \tilde{p}_{q\to i}^{(t-1)}\left(\boldsymbol{\lambda}_q^{[n]}\right)\right|\\
&\le (K-1)N^{K-1}\max_{j,q\in\mathcal{K}\setminus\{k\},n}\left|p_{q\to i}^{(t-1)}\left(\boldsymbol{\lambda}_q^{[n]}\right) - \tilde{p}_{q\to i}^{(t-1)}\left(\boldsymbol{\lambda}_q^{[n]}\right)\right|^{J-1}\\
&\le (K-1)N^{K-1}\max_{j,k,n}\left|p_{k\to i}^{(t-1)}\left(\boldsymbol{\lambda}_k^{[n]}\right) - \tilde{p}_{k\to i}^{(t-1)}\left(\boldsymbol{\lambda}_k^{[n]}\right)\right|\\
&= (K-1)N^{K-1}\left\|\mathcal{M}^{(t-1)} - \widetilde{\mathcal{M}}^{(t-1)}\right\|_{\sup}.
\end{aligned}\tag{28}$$

The inequality (a) in (28) is derived by exploiting the following two facts: 1) $0 < \exp(\mu F_i(\boldsymbol{\Lambda})) \le 1$, since $F_i(\boldsymbol{\Lambda})$ is non-positive and μ is positive, and 2) $\sum_s |x_s| \le |\sum_s (x_s)|$ for arbitrary x_s. The inequality (b) in (28) can be obtained from: 1) the following lemma, and 2) the fact that $\sum_{h\in\mathcal{K}\setminus\{k\}}\sum_{\boldsymbol{\lambda}_h=\boldsymbol{\lambda}_h^{[1]}}^{\boldsymbol{\lambda}_h^{[N]}}$ has to carry out the additions of N^{K-1} items.

Lemma 2: Given $0 \le a_1, \cdots, a_K \le 1$ and $0 \le \tilde{a}_1, \cdots, \tilde{a}_K \le 1$, we have

$$\max_{k\in\mathcal{K}}\left|\prod_{q\in\mathcal{K}\setminus\{k\}} a_q - \prod_{q\in\mathcal{K}\setminus\{k\}}\tilde{a}_q\right| \le (K-1)\max_{q\in\mathcal{K}\setminus\{k\}}|a_q - \tilde{a}_q|. \tag{29}$$

Proof: Please refer to Appendix F.

From (28), we can infer that $\boldsymbol{\Gamma}$ is a continuous mapping, since the coefficient $(K-1)N^{K-1}$ is a constant, and this completes the proof. □

Appendix B
Proof of Theorem 1

Let $\mathcal{S}$ be the collection of the message set $\mathcal{M}^{(t)}$. The mapping function $\boldsymbol{\Theta}$ maps $\mathcal{S}$ to $\mathcal{S}$ with the aid of the function $\boldsymbol{\Gamma}$. According to Lemma 1, $\boldsymbol{\Theta}$ is continuous since $\boldsymbol{\Gamma}$ is continuous. Furthermore, it is clear that the set $\mathcal{S}$ is convex, closed and bounded. Based on Schauder's fixed point theorem, $\boldsymbol{\Theta}$ has a fixed point. This completes the proof. □

Appendix C
Proof of Theorem 2

A. The Average Degree of Factor Nodes

Without a loss of generality, we carry out the analysis for a typical MU located at the origin and assume that the potential serving SBSs are located at the point x_B. The fading (power) is denoted by h_{x_B}, which is assumed to be exponentially distributed, i.e., we have $h_{x_B} \sim \exp(1)$. The path-loss function is given by $\|x_B\|^{-\alpha}$, where $\|\cdot\|$ denotes the Euclidian distance.

The average degree of a factor node in the factor graph is equivalent to the number of SBSs that can provide a high enough SINR ($\ge \delta$) for the typical MU, which can be formulated as

$$N_B = \int_{\mathbb{R}^2}\lambda_B \Pr\left(\rho(x_B) \ge \delta\right)\mathrm{d}x_B, \tag{30}$$

where $\rho(x_B)$ represents the SINR at the typical MU received from the SBSs located at x_B.

We first focus on the probability $\Pr(\rho(x_B) \ge \delta)$ in (30) as follows.

$$\begin{aligned}
\Pr\left(\rho(x_B) \ge \delta\right) &= \Pr\left(\frac{Ph_{x_B}\|x_B\|^{-\alpha}}{\sum\limits_{x_k\in\Phi_B} Ph_{x_k}\|x_k\|^{-\alpha} + \sigma^2} \ge \delta\right)\\
&= \Pr\left(h_{x_B} \ge \frac{\delta(I+\sigma^2)}{P\|x_B\|^{-\alpha}}\right)\\
&= \mathbb{E}_I\left(\exp(-sI)\right)\exp(-s\sigma^2),
\end{aligned}\tag{31}$$

where x_k denotes the location of an interfering SBS, $I \triangleq \sum_{x_k \in \Phi_B} P h_{x_k} \|x_k\|^{-\alpha}$ represents the aggregate interference, and $s = \frac{\delta \|x_B\|^{\alpha}}{P}$. The last step is due to the exponential distribution of h_{x_B}. Then, we derive $\mathbb{E}_I(\exp(-sI))$ in (31) as

$$\begin{aligned}
&\mathbb{E}_I\left(\exp(-sI)\right)\\
&\overset{(a)}{=} \mathbb{E}_{\Phi_B}\left(\prod_{x_k \in \Phi_B} \int_0^{\infty} \exp\left(-sPh_{x_k}\|x_k\|^{-\alpha}\right)\exp\left(-h_{x_k}\right)\mathrm{d}h_{x_k}\right)\\
&\overset{(b)}{=} \exp\left(-\lambda_B \int_{\mathbb{R}^2}\left(1-\frac{1}{1+sP\|x_k\|^{-\alpha}}\right)\mathrm{d}x_k\right)\\
&= \exp\left(-2\pi\lambda_B \frac{1}{\alpha}(sP)^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)\right), \qquad (32)
\end{aligned}$$

where (a) is based on the independence of channel fading, and (b) follows from $\mathbb{E}\left(\prod_x u(x)\right) = \exp(-\lambda \int_{\mathbb{R}^2}(1-u(x))\mathrm{d}x)$, where $x \in \Phi$ and Φ is an PPP in $\mathbb{R}^2$ with the intensity λ [30].

Based on the derivation above, the average degree of the typical MU can be calculated as

$$\begin{aligned}
N_B &= \lambda_B \int_{\mathbb{R}^2} \exp\left(-2\pi\frac{\lambda_B}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)\|x_B\|^2 - \frac{\delta\sigma^2}{P}\|x_B\|^{\alpha}\right)\mathrm{d}x_B\\
&= 2\pi\lambda_B \int_0^{\infty} \exp\left(-2\pi\frac{\lambda_B}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) r^2 - \frac{\delta\sigma^2}{P} r^{\alpha}\right) r\mathrm{d}r. \qquad (33)
\end{aligned}$$

B. The Average Degree of Variable Nodes

In this subsection, we consider a typical SBS which is located at the origin, and assume that an MU is located at the point x_U. The average degree of a variable node in the factor graph is equivalent to the number of MUs that can receive at a high enough SINR ($\geq \delta$) from the typical SBS, which can be formulated as

$$N_U = \int_{\mathbb{R}^2} \lambda_U \Pr\left(\rho(x_U) \geq \delta\right)\mathrm{d}x_U, \qquad (34)$$

where $\rho(x_U)$ represents the received SINR at the MU located at x_U from the typical SBS, i.e.,

$$\begin{aligned}
&\Pr\left(\rho(x_U) \geq \delta\right)\\
&= \Pr\left(\frac{Ph_{x_U}\|x_U\|^{-\alpha}}{\sum_{x_k \in \Phi_B} Ph_{x_k}\|x_k - x_U\|^{-\alpha} + \sigma^2} \geq \delta\right), \qquad (35)
\end{aligned}$$

where x_k denotes the location of an interfering SBS.

Since the PPP is a stationary process, the distribution of $\|x_k - x_U\|$ is independent of the value of x_U, i.e., we have $p(\|x_k - x_U\|) = p(\|x_k\|)$, where $p(\cdot)$ represents the probability density function. Then, we have similar results to Eq. (31). That is, we have

$$\Pr\left(\rho(x_U) > \delta\right) = \mathbb{E}_I\left(\exp(-sI)\right)\exp(-s\sigma^2), \qquad (36)$$

where $s = \frac{\delta\|x_U\|^{\alpha}}{P}$. Then we arrive at

$$N_U = 2\pi\lambda_U \int_0^{\infty} \exp\left(-2\pi\frac{\lambda_B}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) r^2 - \frac{\delta\sigma^2}{P} r^{\alpha}\right) r\mathrm{d}r. \qquad (37)$$

By combining Eqs. (37) and (33), we complete the proof. □

APPENDIX D
PROOF OF COROLLARY 1

When ignoring the noise, we have

$$\begin{aligned}
&Z(\lambda_B, P, \alpha, \delta)\\
&= \int_0^{\infty} \exp\left(-\frac{2\pi\lambda_B}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) r^2\right) r\mathrm{d}r\\
&= \frac{1}{2}\int_0^{\infty} \exp\left(-\lambda_B \frac{2\pi}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right) t\right)\mathrm{d}t\\
&= \frac{1}{2\lambda_B \frac{2\pi}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)} = \frac{\alpha}{4\pi\lambda_B B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)\delta^{\frac{2}{\alpha}}}. \qquad (38)
\end{aligned}$$

By substituting the above expression into (17) and (16), we obtain (20) and (21) respectively. This completes the proof. □

APPENDIX E
PROOF OF THEOREM 3

We conduct the analysis for a typical MU that is located at the origin. We assume that when downloading a file in $\mathcal{F}_n$, the MU will always associate with its nearest SBS, which caches $\mathcal{F}_n$. Note that the OP derived under this assumption is an upper bound for the exact OP. This is because the MU will associate with the second-nearest SBS if it can provide a higher received SINR than that provided by the nearest SBS. Therefore, in some cases, the nearest SBS cannot provide a higher enough SINR ($\geq \delta$), while the second-nearest SBS can. According to our assumption, we will neglect these cases, which leads to a higher OP.

Let us denote by z the distance between the typical MU and the nearest SBS that caches $\mathcal{F}_n$. The location of the nearest SBS caching $\mathcal{F}_n$ is denoted by x_Z. The fading (power) for an SBS located at x_B, $\forall x_B \in \Phi_B$, is denoted by h_{x_B}, which is assumed to be exponentially distributed, i.e., $h_{x_B} \sim \exp(1)$. The path-loss function for a given point x_B is $\|x_B\|^{-\alpha}$.

When random caching is adopted, the distribution of the SBSs that cache $\mathcal{F}_n$ can be modeled as an PPP with the intensity of $\Omega_{\mathcal{F}_n}\lambda_B$. The event that the typical MU can download a file in $\mathcal{F}_n$ from an SBS means that the received SINR from the nearest

SBS which caches $\mathcal{F}_n$ is no less than the threshold δ. Let us denote by $\rho(x_Z)$ the received SINR at the typical MU from the nearest SBS. Then the average probability that the MU can download the file from an SBS is

$$\begin{aligned}
&\Pr\left(\rho(x_Z) \geq \delta\right) \\
&= \int_0^\infty \Pr\left(\frac{h_{x_Z} z^{-\alpha}}{\sum\limits_{x_k \in \Phi_B \setminus \{x_Z\}} h_{x_k} \|x_k\|^{-\alpha}} \geq \delta \,\middle|\, z \right) f_Z(z)\mathrm{d}z \\
&= \int_0^\infty \Pr\left(h_{x_Z} \geq \frac{\delta \left(\sum\limits_{x_k \in \Phi_B \setminus \{x_Z\}} h_{x_k} \|x_k\|^{-\alpha} \right)}{z^{-\alpha}} \,\middle|\, z \right) \\
&\quad \cdot 2\pi \Omega_{\mathcal{F}_n} \lambda_B z \exp\left(-\pi \Omega_{\mathcal{F}_n} \lambda_B z^2\right) \mathrm{d}z \\
&= \int_0^\infty \mathbb{E}_I\left(\exp(-z^\alpha \delta I)\right) 2\pi \Omega_{\mathcal{F}_n} \lambda_B z \exp\left(-\pi \Omega_{\mathcal{F}_n} \lambda_B z^2\right) \mathrm{d}z,
\end{aligned} \tag{39}$$

where we have $I \triangleq \sum\limits_{x_k \in \Phi_B \setminus \{x_Z\}} h_{x_k} \|x_k\|^{-\alpha}$, and the PDF of z, i.e., $f_Z(z)$, is derived by the null probability of a Poisson process with the intensity of $\Omega_{\mathcal{F}_n}\lambda_B$. Note that the interference I consists of I_1 and I_2, where I_1 is emanating from the SBSs caching the FGs $\mathcal{F}_q, \forall\, q \in \boldsymbol{\mathcal{N}}, q \neq n$, while I_2 is from the SBSs caching $\mathcal{F}_n$ excluding x_Z. The SBSs contributing to I_1, denoted by $\Phi_{\bar{n}}$, have the intensity $(1-\Omega_{\mathcal{F}_n})\lambda_B$, while those contributing to I_2, denoted by Φ_n, have the intensity $\Omega_{\mathcal{F}_n}\lambda_B$. Correspondingly, the calculation of $\mathbb{E}_I(\exp(-z^\alpha \delta I))$ will be split into the product of two expectations over I_1 and I_2. The expectation over I_1 directly follows (32), i.e., we have

$$\mathbb{E}_{I_1}\left(\exp(-z^\alpha \delta I_1)\right) = \exp\Big(-\pi \left(1-\Omega_{\mathcal{F}_n}\right) \lambda_B C(\delta,\alpha) z^2\Big), \tag{40}$$

where $C(\delta,\alpha)$ has been defined as $\frac{2}{\alpha}\delta^{\frac{2}{\alpha}} B\left(\frac{2}{\alpha}, 1-\frac{2}{\alpha}\right)$. The expectation over I_2 has to take into account z as the distance from the nearest interfering SBS, i.e., we obtain

$$\begin{aligned}
&\mathbb{E}_{I_2}\left(\exp(-z^\alpha \delta I_2)\right) \\
&= \exp\left(-\Omega_{\mathcal{F}_n}\lambda_B 2\pi \int_z^\infty \left(1 - \frac{1}{1+z^\alpha \delta r^{-\alpha}}\right) r\mathrm{d}r\right) \\
&\overset{(a)}{=} \exp\left(-\Omega_{\mathcal{F}_n}\lambda_B \pi \delta^{\frac{2}{\alpha}} z^2 \frac{2}{\alpha} \int_{\delta^{-1}}^\infty \frac{x^{\frac{2}{\alpha}-1}}{1+x}\mathrm{d}x\right) \\
&\overset{(b)}{=} \exp\left(-\Omega_{\mathcal{F}_n}\lambda_B \pi \delta z^2 \frac{2}{\alpha-2} {}_2F_1\left(1, 1-\frac{2}{\alpha}; 2-\frac{2}{\alpha}; -\delta\right)\right),
\end{aligned} \tag{41}$$

where (a) defines $x \triangleq \delta^{-1} z^{-\alpha} r^\alpha$, and ${}_2F_1(\cdot)$ in (b) is the hypergeometric function. Since we have defined $A(\delta,\alpha) = \frac{2\delta}{\alpha-2} {}_2F_1\left(1, 1-\frac{2}{\alpha}; 2-\frac{2}{\alpha}; -\delta\right)$, by substituting (40) and (41) into (39), we have

$$\begin{aligned}
&\Pr\left(\rho(x_Z) \geq \delta\right) = \int_0^\infty \exp\left(-\pi\left(1-\Omega_{\mathcal{F}_n}\right)\lambda_B C(\delta,\alpha) z^2\right) \\
&\exp\left(-\pi \Omega_{\mathcal{F}_n}\lambda_B z^2 A(\delta,\alpha)\right) 2\pi \Omega_{\mathcal{F}_n}\lambda_B z \exp\left(-\pi \Omega_{\mathcal{F}_n}\lambda_B z^2\right) \mathrm{d}z \\
&= \frac{\Omega_{\mathcal{F}_n}}{C(\delta,\alpha)\left(1-\Omega_{\mathcal{F}_n}\right) + A(\delta,\alpha)\Omega_{\mathcal{F}_n} + \Omega_{\mathcal{F}_n}}.
\end{aligned} \tag{42}$$

It is clear that $\Pr(\mathcal{Q}_n) = 1 - \Pr(\rho(z) \geq \delta)$. This completes the proof. □

Appendix F
Proof of Lemma 2

Without loss of generality, we assume $k=1$. Then (29) becomes

$$\left| \prod_{q=2}^K a_q - \prod_{q=2}^K \tilde{a}_q \right| \leq (K-1) \max_{q \in \{2,\cdots,K\}} |a_q - \tilde{a}_q|. \tag{43}$$

Again, without loss of generality, we assume

$$|a_2 - \tilde{a}_2| \geq \cdots \geq |a_K - \tilde{a}_K|. \tag{44}$$

First, we prove that $|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| \leq 2|a_{K-1} - \tilde{a}_{K-1}|$, under the condition of $|a_{K-1} - \tilde{a}_{K-1}| \geq |a_K - \tilde{a}_K|$. To prove this, we discuss the following possible cases.

1) When $a_{K-1} \geq \tilde{a}_{K-1}$ and $a_K \geq \tilde{a}_K$: We have $a_K \leq a_{K-1} - \tilde{a}_{K-1} + \tilde{a}_K$. Then

$$\begin{aligned}
&|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| \\
&\quad \leq |a_{K-1}(a_{K-1} - \tilde{a}_{K-1} + \tilde{a}_K) - \tilde{a}_{K-1}\tilde{a}_K| \\
&\quad = |(a_{K-1} + \tilde{a}_K)(a_{K-1} - \tilde{a}_{K-1})| \\
&\quad \leq 2|a_{K-1} - \tilde{a}_{K-1}|.
\end{aligned} \tag{45}$$

2) When $a_{K-1} \geq \tilde{a}_{K-1}$, $a_K \leq \tilde{a}_K$, and $a_{K-1}a_K \geq \tilde{a}_{K-1}\tilde{a}_K$: We have

$$\begin{aligned}
|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| &\leq |a_{K-1}\tilde{a}_K - \tilde{a}_{K-1}\tilde{a}_K| \\
&= |a_{K-1} - \tilde{a}_{K-1}|\tilde{a}_K \leq |a_{K-1} - \tilde{a}_{K-1}|.
\end{aligned} \tag{46}$$

3) When $a_{K-1} \geq \tilde{a}_{K-1}$, $a_K \leq \tilde{a}_K$, and $a_{K-1}a_K \leq \tilde{a}_{K-1}\tilde{a}_K$: We have

$$\begin{aligned}
|\tilde{a}_{K-1}\tilde{a}_K - a_{K-1}a_K| &\leq |a_{K-1}\tilde{a}_K - a_{K-1}a_K| \\
&= |a_K - \tilde{a}_K| a_{K-1} \leq |a_{K-1} - \tilde{a}_{K-1}|.
\end{aligned} \tag{47}$$

4) When $a_{K-1} \leq \tilde{a}_{K-1}$, $a_K \geq \tilde{a}_K$, and $a_{K-1}a_K \geq \tilde{a}_{K-1}\tilde{a}_K$: We have

$$\begin{aligned}
|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| &\leq |\tilde{a}_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| \\
&= |a_K - \tilde{a}_K|\tilde{a}_{K-1} \leq |a_{K-1} - \tilde{a}_{K-1}|.
\end{aligned} \tag{48}$$

5) When $a_{K-1} \le \tilde{a}_{K-1}$, $a_K \ge \tilde{a}_K$, and $a_{K-1}a_K \le \tilde{a}_{K-1}\tilde{a}_K$: We have

$$\begin{aligned}|\tilde{a}_{K-1}\tilde{a}_K - a_{K-1}a_K| &\le |\tilde{a}_{K-1}a_K - a_{K-1}a_K| \\ &= |a_{K-1} - \tilde{a}_{K-1}|a_K \le |a_{K-1} - \tilde{a}_{K-1}|. \end{aligned} \quad (49)$$

6) When $a_{K-1} \le \tilde{a}_{K-1}$, $a_K \le \tilde{a}_K$: We have $a_K \ge \tilde{a}_K + a_{K-1} - \tilde{a}_{K-1}$. Then

$$\begin{aligned}|\tilde{a}_{K-1}\tilde{a}_K - a_{K-1}a_K| &\le |\tilde{a}_{K-1}\tilde{a}_K - a_{K-1}(\tilde{a}_K + a_{K-1} - \tilde{a}_{K-1})| \\ &= |(a_{K-1} + \tilde{a}_K)(\tilde{a}_{K-1} - a_{K-1})| \\ &\le 2|a_{K-1} - \tilde{a}_{K-1}|. \end{aligned} \quad (50)$$

From the above discussions, we can see that $|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| \le 2|a_{K-1} - \tilde{a}_{K-1}|$.

Second, as there is $|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| \le 2|a_{K-1} - \tilde{a}_{K-1}|$, we have $|a_{K-1}a_K - \tilde{a}_{K-1}\tilde{a}_K| \le 2|a_{K-2} - \tilde{a}_{K-2}|$. With this condition, we can prove that $|a_{K-2}a_{K-1}a_K - \tilde{a}_{K-2}\tilde{a}_{K-1}\tilde{a}_K| \le 3|a_{K-2} - \tilde{a}_{K-2}|$ by following the similar steps above. By doing this iteratively, we have

$$\left|\prod_{q=2}^{K} a_q - \prod_{q=2}^{K} \tilde{a}_q\right| \le (K-1)|a_2 - \tilde{a}_2|. \quad (51)$$

This completes the proof. □

REFERENCES

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2013–2018," Cisco, San Jose, CA, USA. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf

[2] F. Boccardi *et al.*, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[3] A. Damnjanovic *et al.*, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 10–21, Jun. 2011.

[4] J. Akhtman and L. Hanzo, "Heterogeneous networking: An enabling paradigm for ubiquitous wireless communications," *Proc. IEEE*, vol. 98, no. 2, pp. 135–138, Feb. 2010.

[5] S. Bayat, R. Louie, Z. Han, B. Vucetic, and Y. Li, "Distributed user association and femtocell allocation in heterogeneous wireless networks," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 3027–3043, Aug. 2014.

[6] M. Mirahmadi, A. Al-Dweik, and A. Shami, "Interference modeling and performance evaluation of heterogeneous cellular networks," *IEEE Trans. Commun.*, vol. 62, no. 6, pp. 2132–2144, Jun. 2014.

[7] A. Gupta, H. Dhillon, S. Vishwanath, and J. Andrews, "Downlink multi-antenna heterogeneous cellular network with load balancing," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 4052–4067, Nov. 2014.

[8] Y. Kishiyama, A. Benjebbour, T. Nakamura, and H. Ishii, "Future steps of LTE-A: Evolution toward integration of local area and wide area systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 12–18, Feb. 2013.

[9] T. Nakamura *et al.*, "Trends in small cell enhancements in LTE Advanced," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 98–105, Feb. 2013.

[10] Y. Li, H. Celebi, M. Daneshmand, C. Wang, and W. Zhao, "Energy-efficient femtocell networks: Challenges and opportunities," *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 99–105, Dec. 2013.

[11] N. Golrezaei, A. Molisch, A. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[12] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen, "Coding or not: Optimal mobile data offloading in opportunistic vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 318–333, Feb. 2014.

[13] J. Xu, Q. Hu, W.-C. Lee, and D. Lee, "Performance evaluation of an optimal cache replacement policy for wireless data dissemination," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 125–139, Jan. 2004.

[14] Y. Li *et al.*, "Multiple mobile data offloading through disruption tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1579–1596, Jul. 2014.

[15] D. Chambers, "Data caching reduces backhaul costs for small cells and Wi-Fi," Thinksmallcell, Bath, U.K., Thinksmallcell Forum, Tech. Rep., May 2013.

[16] H. Sarkissian, "The business case for caching in 4G LTE networks," Wireless 2020, Tech. Rep., 11 2014.

[17] "Rethinking the small cell business model," Intel, Santa Clara, CA, USA, Tech. Rep., 2012.

[18] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. 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.

[19] N. Golrezaei, P. Mansourifard, A. Molisch, and A. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[20] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," arXiv preprint arXiv:1305.5216, to be published. AQ1

[21] M. Ji, G. Caire, and A. Molisch, "Optimal throughput-outage trade-off in wireless one-hop caching networks," in *Proc. IEEE ISIT*, Jul. 2013, pp. 1461–1465.

[22] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[23] K. Shanmugam, N. Golrezaei, A. Dimakis, A. 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.

[24] C. C. Moallemi and B. Van Roy, "Resource allocation via message passing," *INFORMS J. Comput.*, vol. 23, no. 2, pp. 205–219, 2011.

[25] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum–product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[26] S. Bavarian and J. Cavers, "Reduced-complexity belief propagation for system-wide MUD in the uplink of cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 3, pp. 541–549, Apr. 2008.

[27] I. Sohn, S. H. Lee, and J. Andrews, "Belief propagation for distributed downlink beamforming in cooperative MIMO cellular networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 12, pp. 4140–4149, Dec. 2011.

[28] D. Stoyan, W. Kendall, and M. Mecke, *Stochastic Geometry and Its Applications*, 2nd ed. New York, NY, USA: Wiley, 2003.

[29] M. Haenggi, J. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1029–1046, Sep. 2009.

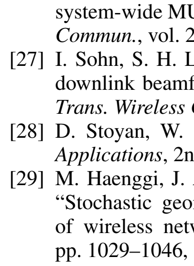
[30] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes, Volume I: Elementary Theory and Methods*. New York, NY, USA: Springer-Verlag, 1996.

[31] H. Dhillon, R. Ganti, F. Baccelli, and J. 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.

[32] S. Rangan and R. Madan, "Belief propagation methods for intercell interference coordination in femtocell networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 631–640, Apr. 2012.

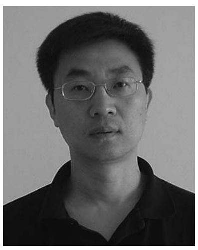
[33] H.-S. Jo, Y. J. Sang, P. Xia, and J. Andrews, "Heterogeneous cellular networks with flexible cell association: A comprehensive downlink SINR analysis," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3484–3495, Oct. 2012.

[34] 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 *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, 2007, pp. 1–14.



Jun Li (M'09) received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009. From January 2009 to June 2009, he was with the Department of Research and Innovation, Alcatel Lucent Shanghai Bell, as a Research Scientist. From June 2009 to April 2012, he was a Postdoctoral Fellow at the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia. Since April 2012, he has been a Research Fellow at the School of Electrical Engineering, The University of Sydney, Sydney, Australia. His research interests include network information theory, channel coding theory, wireless network coding, and cooperative communications. He has served as the Technical Program Committee Member for several international conferences such as GlobeCom2015, ICC2014, VTC2014 (Fall), and ICC2014.

Youjia Chen received the B.S. and M.S degrees in communication engineering from Nanjing University, Nanjing, China, in 2005 and 2008, respectively. She is currently working toward the Ph.D. degree in wireless engineering at The University of Sydney, Sydney, Australia. Her current research interests include resource management, load balancing, and caching strategy in heterogeneous cellular networks.



Zihuai Lin (S'98–M'99–SM'11) received the Ph.D. degree in electrical engineering from Chalmers University of Technology, Gothenburg, Sweden, in 2006. Prior to this, he has held positions at Ericsson Research, Stockholm, Sweden. Following Ph.D. graduation, he was a Research Associate Professor at Aalborg University, Aalborg, Denmark, and currently at the School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia. His research interests include graph theory, source/channel/network coding, coded modulation, MIMO, OFDMA, SC-FDMA, radio resource management, cooperative communications, small-cell networks, 5G cellular systems, etc.

Wen Chen (M'03–SM'11) received the B.S. and M.S. degrees from Wuhan University, Wuhan, China, in 1990 and 1993, respectively, and the Ph.D. degree from the University of Electro-Communications, Tokyo, Japan, in 1999. He was a Researcher at the Japan Society for the Promotion of Science from 1999 through 2001. In 2001, he joined the University of Alberta, Canada, starting as a Postdoctoral Fellow with the Information Research Laboratory and continuing as a Research Associate in the Department of Electrical and Computer Engineering. Since 2006, he has been a Full Professor at the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is also the Director of the Institute for Signal Processing and Systems. His interests cover network coding, cooperative communications, cognitive radio, and MIMO-OFDM systems.

Branka Vucetic (M'83–SM'00–F'03) currently holds the Peter Nicol Russel Chair of Telecommunications Engineering at The University of Sydney, Sydney, Australia. During her career, she has held various research and academic positions in Yugoslavia, Australia, U.K., and China. She has co-authored four books and more than 400 papers in telecommunications journals and conference proceedings. Her research interests include wireless communications, coding, digital communication theory, and machine-to-machine communications. Prof. Vucetic has been elected to the grade of IEEE Fellow for contributions to the theory and applications of channel coding. AQ2

Lajos Hanzo (M'91–SM'92–F'04) received the M.S. degree in electronics and the Ph.D. degree from the Technical University of Budapest, Budapest, Hungary, in 1976 and 1983, respectively; the D.Sc. degree from the University of Southampton, Southampton, U.K., in 2004; and the "Doctor Honoris Causa" degree from the Technical University of Budapest in 2009. During his 38-year career in telecommunications, he has held various research and academic posts in Hungary, Germany, and the U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, where he holds the Chair in Telecommunications. He is currently directing a 60-strong academic research team, working on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council, the European Research Council's Advanced Fellow Grant, and the Royal Society's Wolfson Research Merit Award. During 2008–2012, he was a Chaired Professor at Tsinghua University, Beijing. He is an enthusiastic supporter of industrial and academic liaison and offers a range of industrial courses. He has successfully supervised about 100 Ph.D. students, coauthored 20 John Wiley/IEEE Press books on mobile radio communications totaling in excess of 10 000 pages, and published more than 1500 research entries at IEEE Xplore. His research is funded by the European Research Council's Senior Research Fellow Grant. Dr. Hanzo is a Fellow of the Royal Academy of Engineering, the Institution of Engineering and Technology, and the European Association for Signal Processing. He is also a Governor of the IEEE Vehicular Technology Society. During 2008–2012, he was the Editor-in-Chief of IEEE Press. He has served as the Technical Program Committee Chair and the General Chair of IEEE conferences, has presented keynote lectures, and has been awarded a number of distinctions. He has more than 22 000 citations. For further information on research in progress and associated publications, please refer to http://www-mobile.ecs.soton.ac.uk

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

AQ1 = Please provide publication update in Ref. [20].
AQ2 = Please provide details on the educational background of author Branka Vucetic.

END OF ALL QUERIES