A Mean Field Game-Based Distributed Edge Caching in Fog Radio Access Networks

Yanxiang Jiang, *Senior Member, IEEE*, Yabai Hu, Mehdi Bennis, *Senior Member, IEEE*,
Fu-Chun Zheng, *Senior Member, IEEE*, and Xiaohu You, *Fellow, IEEE*

***Abstract*—In this paper, the edge caching optimization problem in fog radio access networks (F-RANs) is investigated. Taking into account time-variant user requests and ultra-dense deployment of fog access points (F-APs), we propose a distributed edge caching scheme to jointly minimize the request service delay and fronthaul traffic load. Considering the interactive relationship among F-APs, we model the optimization problem as a stochastic differential game (SDG) which captures the dynamics of F-AP states. To address both the intractability problem of the SDG and the caching capacity constraint, we propose to solve the optimization problem in a distributive manner. Firstly, a mean field game (MFG) is converted from the original SDG by exploiting the ultra-dense property of F-RANs, and the states of all F-APs are characterized by a mean field distribution. Then, an iterative algorithm is developed that enables each F-AP to obtain the mean field equilibrium and caching control without extra information exchange with other F-APs. Secondly, a fractional knapsack problem is formulated based on the mean field equilibrium, and a greedy algorithm is developed that enables each F-AP to obtain the final caching policy subject to the caching capacity constraint. Simulation results show that the proposed scheme outperforms the baselines.**

***Index Terms*—Fog radio access networks, distributed edge caching, mean field game, fractional knapsack problem, request service delay, fronthaul traffic load.**

Manuscript received May 20, 2019, revised September 1, 2019, October 31, 2019, and accepted December 11, 2019. This work was supported in part by the Natural Science Foundation of China under Grant 61971129, the Natural Science Foundation of Jiangsu Province under Grant BK20181264, the National Key R&D Program of China under Grant 2018YFB1801103, the Research Fund of the State Key Laboratory of Integrated Services Networks (Xidian University) under Grant ISN19-10, the Research Fund of the Key Laboratory of Wireless Sensor Network & Communication (Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences) under Grant 2017002, and the Research Fund of the School of Electronic and Information Engineering, Harbin Institute of Technology (Shenzhen) under Grant HITSZ20190631. Part of this work has been presented at the IEEE VTC 2018 FALL, Chicago, USA, August 2018. The corresponding authors of this paper are Y. Jiang and F. Zheng.

Y. Jiang is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China, the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China, and the Key Laboratory of Wireless Sensor Network & Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, 865 Changning Road, Shanghai 200050, China (e-mail: yxjiang@seu.edu.cn).

Y. Hu and X. You are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: xhyu@seu.edu.cn).

M. Bennis is with the Centre for Wireless Communications, University of Oulu, Oulu 90014, Finland (e-mail: mehdi.bennis@oulu.fi).

F. Zheng is with the School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China, and the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China. (e-mail: fzheng@ieee.org).

I. Introduction

The increasing number of mobile devices and surging user demand for multi-media service have triggered massive amounts of data in wireless networks, which brings about huge pressure on fronthaul links. Fog radio access networks (F-RANs) can address this problem by allocating caching and computing resources to fog access points (F-APs) that are deployed ultra-densely at the network edge. In ultra-dense F-RANs, a large deployment density enables F-APs to provide large coverage area and high quality wireless links, while edge caching alleviates fronthaul traffic bottleneck and decreases request service delay by bringing contents closer to users [1]–[3]. Therefore, the system cost on the high-speed fronthaul links can be controlled and experience of users can be improved.

Due to the limited caching capacity of F-APs, the optimization of content placement is a major issue in F-RANs. The content placement design can be centralized or distributed. In centralized caching schemes, caching policies are generally determined by a control center that collects information of the entire network for seeking a global optimal solution. However, the difficulty in information acquisition and global optimization renders centralized schemes impractical in ultra-dense networks. In distributed caching schemes, on the contrary, edge nodes can optimize their caching policies based on local information and certain amounts of information exchange among nodes. Without the need of a central controller, the caching policies can be obtained with a considerable reduction of communication overhead and computation complexity. In [4] and [5], distributed content placement methods were put forward based on belief propagation (BP) algorithm, where each base station (BS) computed the caching policies through iterative message passing between nodes till convergence, and different factor graphs were formulated according to the corresponding network structure to minimize the downloading delay. In [6] and [7], distributed algorithms were implemented based on alternating direction method of multipliers (ADMM). In [6], the cost of fronthaul usage was minimized in mobile cellular networks, whereas in [7] a comprehensive network gain was maximized by jointly taking caching, computing and bandwidth resource allocation into consideration. The authors in [8] modeled the network dynamics in a game-theoretic approach and proposed a distributed caching scheme to minimize the backhaul traffic load. In [9], a Markov-chain-based optimization framework was established to minimize the macro-cell load by considering a high-mobility scenario, and the original problem was decomposed into independent subproblems that could be solved in a distributed manner. In [10], the authors proposed a distributed cache replacement strategy based on Q-learning in device-to-device (D2D)-enabled networks, where reward function of each BS was computed according to its local statistics in previous time periods. In [11], a coded caching scheme in F-RANs was proposed by considering asynchronous requests, where F-APs randomly prefetch certain bits of each file in the placement phase in a decentralized fashion. However, all the above works are based on the assumption of a static content popularity distribution, whereas user requests for popular contents generally change over time in practice.

By considering time-varying characteristics of user requests, only a few works have focused on the design of dynamic distributed edge caching schemes. In [12] and [13], the dynamic evolution of content popularity was investigated by using differential equations and Markov chains, respectively. In [14], a distributed gradient ascent algorithm was proposed to resiliently maximize the latency reduction in an Internet cache topology. In [15], the authors considered a fairness problem in a data-sharing network, and proposed a distributed algorithm that can continuously select nodes to cache contents adaptive to newly generated data. In [16], the content demands of users were depicted by a queue system, and the optimization problem was solved by a Lyapunov-based algorithm. The caching scheme proposed in [16] jointly optimized data forwarding and content caching in a wireless network. To retrieve the demand and congestion information over the network, the authors in [16] established a novel dynamic mapping in a dual queue system, which enables each node to obtain the related information based only on the local queue. However, the proposed scheme in [16] requires each caching node to observe the caching state of adjacent nodes, which incurs extra communications overhead and delay. Consequently, it may not be suitable in ultra-dense F-RANs. Although our (proposed) distributed caching scheme does not take the network congestion and data forwarding issue into consideration, it is able to avoid extra communications overhead among adjacent F-APs by exploiting the density of F-AP deployment, which is more suitable in ultra-dense F-RANs. Nevertheless, most of the above works require considerable amounts of communications among edge nodes, and the induced overhead is intolerable in ultra-dense networks. Moreover, none of the above works account for both request service delay and fronthaul traffic load in radio access networks, which are important performance metrics that affect the quality of experience (QoE) on user sides and the operational expenditures (OPEX) on network sides, respectively [17], [18].

Motivated by the aforementioned discussions, we propose a distributed edge caching scheme adaptive to dynamic user requests in ultra-dense F-RANs, where each F-AP can optimize both metrics of request service delay and fronthaul traffic load in a distributive manner. The main contributions of our paper are summarized below.

1) To jointly minimize the request service delay and fronthaul traffic load, a stochastic differential game (SDG) is formulated to model the edge caching optimization problem with regard to each file. The SDG captures the interactions among F-APs and the dynamics of F-AP states by incorporating the status of time-variant user requests.
2) To reduce the computational complexity of the problem-solving, the SDG is transformed to a mean field game (MFG) by utilizing the ultra-dense property of F-RANs. In the MFG, the individual states of all the F-APs are approximated by a statistical mean field distribution. Each F-AP can then obtain the mean field equilibrium and determine the dynamic caching control of each file based on its local state and the mean field distribution without extra information exchange with the cloud server or adjacent F-APs.
3) To obtain the final optimal caching policy of each file and take into consideration the capacity constraint of F-APs, a fractional knapsack problem is formulated based on the mean field equilibrium and is solved by a greedy algorithm. The final optimal caching policy of each file then can be obtained in a distributive manner.
4) The proposed caching scheme is verified through simulations. Simulation results show that the proposed caching scheme can adapt to time-variant user requests to reduce both the request service delay and fronthaul traffic load. Simulation results also show the superior performance of the proposed caching scheme in comparison with several traditional caching schemes under both static and dynamic user requests.

The rest of the paper is organized as follows. In Section II, the system model is briefly described. In Section III, the edge caching optimization problem is formulated. In Section IV, the proposed distributed edge caching scheme is presented. Simulation results are shown in Section V and final conclusions are drawn in Section VI.

II. System Model

A. Network Model

Consider an edge caching enabled F-RAN consisting of I F-APs, K users, and one remote cloud server as illustrated in Fig. 1, where the F-APs are ultra-densely deployed and connected with the cloud server via fronthaul links. The F-AP set and the user set are denoted by $\mathcal{I} = \{1, 2, \ldots, i, \ldots, I\}$ and $\mathcal{K} = \{1, 2, \ldots, k, \ldots, K\}$, respectively. Consider a time-slotted system, and let $g_{i,k}(t)$ denote the channel fading coefficient between F-AP i and user k in time slot t. By considering a time-varying channel, the evolution of $g_{i,k}(t)$ can be characterized by a mean-reverting Ornstein-Uhlenbeck process as follows [19]–[21]:

$$\mathrm{d}g_{i,k}(t) = \frac{1}{2}\varsigma\left(\mu_g - g_{i,k}(t)\right)\mathrm{d}t + \sigma_g \mathrm{d}B_{i,k}(t), \qquad (1)$$

where ς, μ_g and σ_g are positive real numbers, μ_g is the long-term mean of the process and ς depicts the speed of the process towards the long-term mean, $\frac{\sigma_g^2}{\varsigma}$ is the long-term variance, and $B_{i,k}(t)$ is a standard Brownian motion. The Ornstein-Uhlenbeck process combines both time-variant

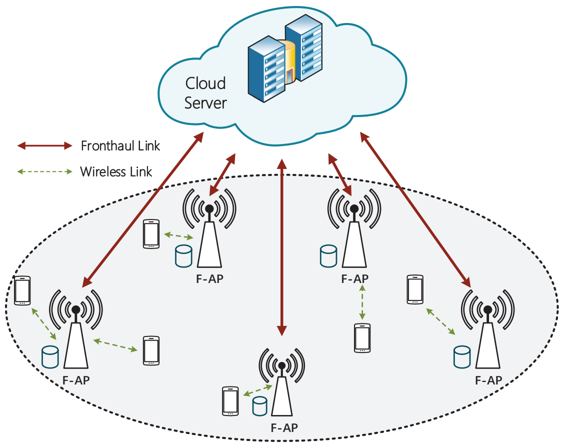


Fig. 1. Illustration of the considered F-RAN.

small-scale and large-scale fading effects: the stochastic term describes the small-scale fading such as multipath fading, while the deterministic term that controls the long-term mean describes the large-scale fading such as shadowing effect. Let $|h_{i,k}(t)|^2$ denote the channel gain between F-AP i and user k in time slot t. Assuming a fixed channel pathloss,[1] then combining the channel fading and path loss, we have $|h_{i,k}(t)|^2 = |g_{i,k}(t)|^2 d_{i,k}^{-\alpha}$, where $d_{i,k}$ denotes the Euclidean distance between F-AP i and user k, and α denotes the pathloss exponent with $\alpha \geq 2$.

Assume that the F-APs and users are uniformly distributed, and each user associates with a default serving F-AP which is the geographically nearest one [22]. Then, all the users, that are associated with the same default serving F-AP i, construct the served user cluster $\mathcal{U}^i$ with respect to the default serving F-AP i, and each user is served by one F-AP at a time. Assume that users request files from the content library $\mathcal{N} = \{1, 2, \ldots, n, \ldots, N\}$. Without loss of generality, each file has the same size of S bits.[2] The request arrival in consecutive time slots is modeled as a Poisson process with the average arrival rate λ per slot. Users get the requested files from the corresponding F-APs through wireless links. Then, the wireless transmission rate from F-AP i to user k in time slot t can be expressed as follows:

$$R_{i,k}(t) = W\log_2\left(1 + \frac{|h_{i,k}(t)|^2 P_i}{\sigma^2 + \sum_{j \neq i, j \in \mathcal{I}} |h_{j,k}(t)|^2 P_i}\right), \quad (2)$$

where W denotes the transmission bandwidth, P_i the transmission power of F-AP i, and σ^2 the noise power. Note that we have assumed that an F-AP only serves one user at a given time, thus simultaneous bandwidth sharing among multiple users is not taken into consideration. The emphasis of this paper is edge caching resource allocation, therefore the power and bandwidth allocation is not the focus of our research, which is the reason why we made the above assumption. Note that our work can be readily extended to multiple user scenarios by employing bandwidth and power allocation schemes.

B. Content Placement and Delivery Model

By considering the content processing policy of each F-AP, each time slot can be divided into two phases: content placement and content delivery. The cloud server stores all the files in the content library, while each F-AP is equipped with a limited caching capacity that can store up to C bits.

In the content placement phase, F-AP i fetches certain fractions of files through the fronthaul link according to its caching policy. Let the N-dimensional vector $\boldsymbol{c}_i(t) = [c_{i,1}(t), c_{i,2}(t), \ldots, c_{i,n}(t), \ldots, c_{i,N}(t)]^{\mathrm{T}}$ denote the caching policy of F-AP i, where $c_{i,n}(t) \in [0,1]$ denotes the instantaneous caching rate of file n and represents the proportion of file n to be cached by prefetching through the fronthaul link in time slot t. Let the N-dimensional vector $\boldsymbol{s}_i(t) = [s_{i,1}(t), s_{i,2}(t), \ldots, s_{i,n}(t), \ldots, s_{i,N}(t)]^{\mathrm{T}}$ denote the caching state of F-AP i, where $s_{i,n}(t) \in [0, S]$ denotes the instantaneous number of bits of file n stored in its caching storage in time slot t. File n will be discarded by F-AP i when it is rarely requested. The less requests for file n, the faster the file to be discarded from the caching storage. Correspondingly, the dynamics of the caching state can be expressed as follows:

$$\mathrm{d}s_{i,n}(t) = S\left(c_{i,n}(t) - e^{a-1}a^{q_{i,n}(t)}\right)\mathrm{d}t, \quad (3)$$

where a denotes a predefined parameter, and $q_{i,n}(t)$ denotes the number of requests for file n from the served user cluster of F-AP i in time slot t. Note that the incrementing rate of the caching state is represented by the caching rate $c_{i,n}(t)$, while the decrementing rate is represented by $e^{a-1}a^{q_{i,n}(t)}$, which is a descending function of $q_{i,n}(t)$, with the parameter $a \in (0,1)$ that controls the steepness of the function. Therefore, a smaller request number for file n results in a larger discarding rate, which adapts to real-time requests. Note that the content removal takes place in each time slot. The removal rate is the decrementing rate of the caching state in equation (3), and it is a descending function of the file request number. Therefore, the less a file is requested, the faster it will be discarded. Moreover, the removed fraction of non-request files in each time slot is the decrementing rate $e^{a-1}a^{q_{i,n}(t)}$. As for the optimization of the constant parameter a with respect to the decrementing rate, when it is set to a small value, the decrementing rate decreases with the request number $q_{i,n}$ very fast. Consequently, only the file with an extremely small request number will be removed from the cache, which results in a slow discarding process. On the contrary, when a is set to a large value, the decrementing rate decreases with the request number $q_{i,n}$ relatively slowly. Consequently, the F-AP tends to discard files at a relatively faster rate. We set the parameter a to 0.1 by tuning through simulations, but have not specifically implemented the optimization with regard to a, which will be an interesting research issue in our future work.

In the content delivery phase, each F-AP processes the user requests during the current time slot according to the caching

[1] Note that the pathloss is assumed to be a constant, which is true within a sufficiently short time frame - especially for the usual scenario of low mobility in F-RANs.

[2] Note that contents with different sizes can always be split into data segments of the same size, and each data segment can then be considered as a content. This is a common practice in real world systems. Here we follow the same assumption and justification as in [23].

states of its own and the other F-APs, which can be classified into three cases. For case 1, a user that requests for file n associates with its default serving F-AP if the file is already cached. Otherwise, for case 2, the user seeks other adjacent F-APs that have cached file n and associates with the one that provides the largest transmission rate [4] that is no less than R_{th}. For case 3, when none of the F-APs has cached the requested file or can provide satisfactory transmission rate, the requesting user will then still associate with the default serving F-AP. Considering that the caching state $s_{i,n}(t)$ is a continuous variable ranging from 0 to S, here we assume that file n is thought to be cached by F-AP i as long as $s_{i,n}(t) \geq \frac{1}{2}S$. Due to the continuity of cache state of each file, there might be some files that have not been cached wholly but with a relatively large portion (e.g., 80%). In this case, it induces relatively small cost to fetch the remaining portion (20%) of the file via fronthaul link, which can be considered by the user as having cached the corresponding file. Therefore, we intuitively set the threshold to 1/2. When the cache state of a file is larger than half the file size at the default serving F-AP, the user will associate with the default serving F-AP; otherwise, the user will search for adjacent F-APs that have cached the file with a proportion no less than 1/2. It would be an interesting issue to tune or optimize this parameter for better caching performance in our future work. Let C_1, C_2 and C_3 denote the caching state conditions for the three cases, respectively. Then, they can be expressed as follows:

$$C_1\left(t, s_{i,n}(t)\right) = \tilde{H}\left(s_{i,n}(t) - \frac{1}{2}S\right), \tag{4}$$

$$C_2\left(t, s_{i,n}(t), s_{-i,n}(t)\right) = \tilde{H}\left(\frac{1}{2}S - s_{i,n}(t)\right) \times \tilde{H}\left(s_{-i,n}(t) - \frac{1}{2}S\right), \tag{5}$$

$$C_3\left(t, s_{i,n}(t), s_{-i,n}(t)\right) = \tilde{H}\left(\frac{1}{2}S - s_{i,n}(t)\right) \times \tilde{H}\left(\frac{1}{2}S - s_{-i,n}(t)\right), \tag{6}$$

where $s_{-i,n}(t)$ denotes the caching state of the other possbile serving F-AP in case 2, $\tilde{H}(\cdot)$ is a smooth approximation of the heaviside step function to guarantee the continuity of the function with $\tilde{H}(x) = 1/(1+e^{-2lx})$ for $l > 0$. Since $\tilde{H}(\cdot)$ is the approximated heaviside step function, C_1 takes the value of 1 when the cache state of F-AP i is larger than half the file size, which corresponds to case 1 where F-AP i has cached file n. C_1 takes the value of 0 otherwise. C_2 takes the value of 1 when the cache state of F-AP i is smaller than half the file size and the cache state of the other adjacent F-AP is larger than half the file size, which corresponds to case 2 where F-AP i has not cached file n and the other adjacent F-AP has cached file n. C_2 takes the value of 0 otherwise. C_3 takes the value of 1 when the cache states of F-AP i and the other adjacent F-APs are all smaller than half the file size, which corresponds to case 3 where F-AP i and the other adjacent F-APs have not cached file n. C_3 takes the value of 0 otherwise. Let $y_{i,k}(t) \in \{0,1\}$ denote the association status of F-AP i with user k, where $y_{i,k}(t) = 1$ if user k is associated with F-AP i, and $y_{i,k}(t) = 0$ otherwise. The associated F-AP serves the requesting user by firstly retrieving the uncached file fraction from the cloud server via the fronthaul link and then transmitting it to the user via the wireless link.

Given the above introduced system model, the state of F-AP i with respect to (w.r.t.) file n can be represented by its channel state in (1) and its caching state in (3). Let $\boldsymbol{g}_i = \left[g_{i,1}, g_{i,2}, \ldots, g_{i,k}, \ldots, g_{i,K}\right]^{\mathrm{T}}$ denote the channel state vector of F-AP i. Then, the state vector of F-AP i w.r.t. file n can be denoted by $\boldsymbol{x}_{i,n} = \left(\boldsymbol{g}_i^{\mathrm{T}}, s_{i,n}\right)^{\mathrm{T}}$.

As stated previously, the service delay of user requests and the cost on high-speed fronthaul links greatly affect user QoE and network expenditure, respectively. Therefore, from the standpoint of QoE improvement and network expenditure reduction, the objective of this paper is to determine the optimal edge caching policy, which can jointly minimize the request service delay and fronthaul traffic load for each F-AP.

III. Problem Formulation

In this Section, we firstly define the cost function for each F-AP, and then formulate the stochastic differential game to model the edge caching optimization problem.

A. Cost Function

Let $J_{i,n}(t)$ denote the total cost of F-AP i w.r.t. file n in time slot t, which is composed of the request service delay and fronthaul traffic load.

1) Request Service Delay: Let $D_{i,n}(t)$ denote the delay cost of F-AP i with respect to file n, which is the total service delay of the requests for file n from its served user cluster $\mathcal{U}^i$ in time slot t. Note that the request service delay refers to the time duration from the file being requested to the user being successfully served, which is composed of the wireless transmission delay and possible fronthaul retrieving delay.

The request service delay can be expressed in (7) as shown at the bottom of this page, where $\mathrm{req}_k(t)$ denotes the requested file of user k in time slot t, R^{F} denotes the fronthaul transmission rate, $s_{-i,n}(t)$ and $R_{-i,k}(t)$ denote the caching

$$D_{i,n}(t) = \sum_{\substack{k \in \mathcal{U}^i \\ \mathrm{req}_k(t)=n}} \left\{ \left[C_1\left(t, s_{i,n}(t)\right) + C_3\left(t, s_{i,n}(t), s_{-i,n}(t)\right)\right]\left[\frac{S}{R_{i,k}(t)} + \frac{S - s_{i,n}(t)}{R^{\mathrm{F}}}\right] + C_2\left(t, s_{i,n}(t), s_{-i,n}(t)\right)\left[\frac{S}{R_{-i,k}(t)} + \frac{S - s_{-i,n}(t)}{R^{\mathrm{F}}}\right]\right\}, \tag{7}$$

state and the transmission rate provided by the other possible serving F-AP in case 2.

The first term in (7) denotes the delay when the requesting user is served by its default F-AP i, which corresponds to case 1 and 3. It is the product of the caching state condition and the corresponding delay under that condition, and refers to the wireless transmission delay between F-AP i and user k and the fronthaul delay induced by retrieving the uncached segment of file n from the cloud server. The second term in (7) denotes the delay when the user is served by certain adjacent F-AP in case 2, and refers to the wireless transmission delay and fronthaul retrieving delay from the other serving F-AP. According to the user association policy introduced in Section II-B, the other possible serving F-APs have to provide both satisfactory transmission rate and at least half the file. Hence, the set of the other possible serving F-APs can be expressed as $\psi_k\left(t\right) = \left\{j\left|R_{j,k}\left(t\right) \geq R_{\mathrm{th}}, s_{j,n}\left(t\right) \geq \frac{S}{2}\right.\right\}$. The associated other serving F-AP for user k in case 2 is the one that provides the largest transmission rate, whose F-AP index can be expressed as follows:

$$-i_{k,n}\left(t\right) = \begin{cases} 0, & \psi_k\left(t\right) = \varnothing, \\ \underset{j\neq i, j\in\psi_k(t)}{\arg\max}\ R_{j,k}\left(t\right), & \text{otherwise.} \end{cases} \tag{8}$$

If the F-AP index equals to 0, then there is no actual other F-AP that satisfies the association requirements. Then, we let $s_{0,n}\left(t\right) = 0$. Note that $s_{-i,n}\left(t\right)$ and $R_{-i,k}\left(t\right)$ can be approximated by the average states of all the other F-APs denoted by $\bar{s}_{-i,n}\left(t\right)$ and $\bar{g}_{-i,k}\left(t\right)$, which will be presented in Section IV-B.

2) Fronthaul Traffic Load: The traffic load on the fronthual link is induced by both content caching in the placement phase and content retrieving in the delivery phase. Let $O_{i,n}\left(t\right)$ denote the fronthaul load of F-AP i with respect to file n in time slot t, and it can be expressed in (9) as shown at the bottom of this page. The first term in (9) characterizes the instantaneous fronthaul load in the content caching process with η_1 and η_2 being the control parameters, and it is a simple quadratic function that is monotonously increasing with the caching rate $c_{i,n}\left(t\right)$. The second term in (9) is the fronthaul load generated by content retrieving when F-AP i has not cached the entire file n in its caching storage. In this case, F-AP i has to repeatedly retrieve the uncached file segment through the fronthual link for each request. Correspondingly, the fronthaul load is the product of the request number, the caching state condition and the size of the file segment to be retrieved. The requests may arise from users both inside and outside the default serving cluster $\mathcal{U}^i$, respectively. For the requesting users inside $\mathcal{U}^i$, the fronthaul traffic load for their requests is generated only when they are served by F-AP i, i.e., in case 1 and case 3. For the requesting users outside $\mathcal{U}^i$, the number of requests can be expressed as $q^-_{i,n}\left(t\right) = \left|\left\{k|k \notin \mathcal{U}^i, \mathrm{req}_k\left(t\right) = n, y_{i,k}\left(t\right) = 1\right\}\right|$, and the fronthaul traffic load is generated when the requested file is not cached in their default serving F-APs, but has been cached in F-AP i. Let $C_4\left(t, s_{i,n}\left(t\right), s_{-i,n}\left(t\right)\right)$ denote the caching state condition for the above case. Then, it can be expressed as follows:

$$\begin{aligned} &C_4\left(t, s_{i,n}\left(t\right), s_{-i,n}\left(t\right)\right) = \\ &\qquad \tilde{H}\left(s_{i,n}\left(t\right) - \frac{1}{2}S\right)\tilde{H}\left(\frac{1}{2}S - s_{-i,n}\left(t\right)\right). \end{aligned} \tag{10}$$

According to (7) and (9), the total instantaneous cost $J_{i,n}\left(t\right)$ of F-AP i w.r.t. file n can be expressed by the weighted summation of the request service delay and fronthaul traffic load as follows:

$$J_{i,n}\left(t\right) = \omega_1 D_{i,n}\left(t\right) + \omega_2 O_{i,n}\left(t\right), \tag{11}$$

where ω_1 and ω_2 denote the weighted coefficients to bring the two corresponding terms to the same scale. Each F-AP seeks the caching policy for each file to minimize the total cost, and the corresponding optimization problem will be formulated in the next subsection.

B. Stochastic Differential Game Formulation

Consider the edge caching optimization problem in a finite time horizon T. As can be observed from (7) and (9), the cost function of each F-AP is determined by the states of all the F-APs. Correspondingly, the states and policies of the F-APs have direct impacts on each other. Therefore, the edge caching optimization problem of the F-APs can be characterized by a stochastic differential game that captures the interactive relationship among them. Note that for the edge caching optimization problem, the limited caching capacity of each F-AP indicates a hard-constrained caching state space for the set of all files. However, sophisticated state and control constraints in a stochastic differential game induce system discontinuity, which renders the analysis of equilibrium quite difficult, especially for the state with a high dimension [24]. To circumvent this issue, we firstly formulate separate stochastic differential games for different files, and the capacity constraint will be considered after we obtain the solution of each game.

The stochastic differential game for file n is defined by a 4-tuple $\left(\mathcal{I}, \left\{\mathcal{X}_{i,n}\right\}_{i\in\mathcal{I}}, \left\{\mathcal{C}_{i,n}\right\}_{i\in\mathcal{I}}, \left\{\mathcal{J}_{i,n}\right\}_{i\in\mathcal{I}}\right)$. $\mathcal{I}$ denotes the set of players, which are F-APs; $\mathcal{X}_{i,n}$ denotes the state space of player i, which includes all the possible state vector $\boldsymbol{x}_{i,n}$; $\mathcal{C}_{i,n}$ denotes the action set of player i, which includes all the possible caching policy $c_{i,n}$; $\mathcal{J}_{i,n}$ denotes the cumulative cost of player i w.r.t file n over the finite time horizon T with $\mathcal{J}_{i,n} = \mathbb{E}\left[\int_0^T J_{i,n}\left(t\right)\mathrm{d}t\right]$.

$$\begin{aligned} O_{i,n}\left(t\right) = &\left(\eta_1 c_{i,n}\left(t\right) + \frac{1}{2}\eta_2 c^2_{i,n}\left(t\right)\right) + \Big[q_{i,n}\left(t\right)\left(C_1\left(t, s_{i,n}\left(t\right)\right) + C_3\left(t, s_{i,n}\left(t\right), s_{-i,n}\left(t\right)\right)\right) \\ &+ q^-_{i,n}(t) C_4\left(t, s_{i,n}\left(t\right), s_{-i,n}\left(t\right)\right)\Big]\left(S - s_{i,n}\left(t\right)\right). \end{aligned} \tag{9}$$

In the above established SDG, the edge caching optimization problem of player i can be formulated as follows:

$$\min_{c_{i,n}(0\to T)} \mathcal{J}_{i,n}\left(c_{i,n}, \boldsymbol{x}_{i,n}, \boldsymbol{x}_{-i,n}\right), \tag{12}$$

where $c_{i,n}\left(0\to T\right)$ denotes the caching policy over the finite time horizon T, and $\boldsymbol{x}_{-i,n} = \{\boldsymbol{x}_{j,n}, j\neq i,\ \ j\in\mathcal{I}\}$ denotes the states of the other $I-1$ players.

According to the theory of dynamic programming, the overall optimum of a problem for the entire time horizon $[0,T]$ can be obtained by sequentially constructing the optimal policies of its subproblems over time period $[t,T]$ in a time-reversed manner [25]. Correspondingly, we can set the value function $v_{i,n}\left(t,\boldsymbol{x}_{i,n}(t)\right)$ of player i w.r.t. file n in time slot t as the minimal cost over time period $[t,T]$ as follows:

$$v_{i,n}\left(t,\boldsymbol{x}_{i,n}(t)\right) = \min_{c_{i,n}(t\to T)} \mathbb{E}\left[\int_t^T J_{i,n}\left(u\right)\mathrm{d}u\right]. \tag{13}$$

It can be observed from the above equation that $v_{i,n}\left(T,\cdot\right)=0$. Given this final value of $v_{i,n}$, the minimal accumulative cost over $[0,T]$ can be computed backwardly.

The solution of the formulated SDG is the Nash equilibrium (NE) [26] that is defined below.

Definition 1 (NE of the SDG): In the SDG, the set of caching policies

$$\left\{c^*_{i,n}\left(0\to T\right) = \left\{\phi^*_{i,n}\left(t,\boldsymbol{x}_{i,n}(t)\right)\right\}_{t=0}^{T}, i\in\mathcal{I}\right\}, \tag{14}$$

where $\phi^*_{i,n}\left(t,\boldsymbol{x}_{i,n}(t)\right)$ denotes the optimal caching control, constitutes a feedback NE, and $\left\{\boldsymbol{x}^*_{i,n}(t),\ \boldsymbol{x}^*_{-i,n}(t), t\in[0,T]\right\}$ is the corresponding state trajectory if the following inequality holds:

$$\mathcal{J}_{i,n}\left(c^*_{i,n}, \boldsymbol{x}^*_{i,n}, \boldsymbol{x}^*_{-i,n}\right) \leq \mathcal{J}_{i,n}\left(c_{i,n}, \boldsymbol{x}^*_{i,n}, \boldsymbol{x}^*_{-i,n}\right), \forall i\in\mathcal{I}. \tag{15}$$

The NE of the SDG can be guaranteed if there exist value functions satisfying the following backward Hamilton-Jacobi-Bellman (HJB) partial differential equations (PDEs) for all $i\in\mathcal{I}$ [26]:

$$\partial_t v_{i,n}\left(t,\boldsymbol{x}_{i,n}(t)\right) + \min_{c_{i,n}(t)}\left[L v_{i,n}\left(t,\boldsymbol{x}_{i,n}(t)\right) + J_{i,n}\left(t\right)\right] = 0, \tag{16}$$

where L denotes the partial differential operator. The optimal caching control for each file can be obtained by solving the I HJBs that are coupled due to the interactions among the F-APs.

To solve the coupled I HJBs for each file, it however requires each F-AP to acquire the state information of all the other F-APs, which requires high computational load and induces enormous extra communicating overhead. Moreover, as stated previously, the edge caching optimization problem of each file is characterized as an independent SDG to avoid the discontinuity. Correspondingly, the total cost of each file is minimized separately, whereas the caching capacity that constrains the fractions of all the cached files is not considered. Therefore, we propose to solve the above two challenging problems by introducing our proposed distributed edge caching scheme in the next Section.

IV. Proposed Distributed Edge Caching Scheme

The formulated SDG in Section III is difficult to solve directly and does not take into consideration the caching capacity constraint. To tackle these two problems, we propose a distributed edge caching scheme by formulating and solving mean field games and fractional knapsack problems. In this section, firstly the general framework of the proposed distributed caching scheme is introduced. Then, the mean field game and the fractional knapsack problem are formulated and solved respectively. Finally, the optimality and complexity of the proposed scheme is analysed.

A. Scheme Description

In the proposed distributed edge caching scheme, the intractability of the solution to the original SDG is addressed by a conversion of the original game to a MFG, in which individual states of all the players can be captured by a single statistical distribution. To obtain the Nash equilibrium of the MFG for each file, an iterative algorithm that involves solving two coupled PDEs is proposed, where each F-AP is able to compute the caching control for each file in a distributed manner with low computational complexity. Then, a fractional knapsack problem is formulated in each time slot to describe the corresponding optimization problem constrained by the caching capacity. Based on the value function and caching control of each file calculated in the mean field game, we propose to solve the fractional knapsack problem in each time slot using a greedy algorithm. Finally, the state trajectory and final optimal caching policy of each file in each time slot can be obtained.

By implementing the iterative algorithm and greedy algorithm, each F-AP can finally obtain the optimal caching policy of each file satisfying the caching capacity constraint. The number of PDEs for obtaining the caching control in the MFG has been reduced from primarily $I\times N$ to $2\times N$. However, when the content library is considerably large, it still needs high computational complexity for each F-AP to solve the $2\times N$ PDEs. In fact, even the library size N is considerably large, the request probability for most files is still considerably small. Furthermore, the files that need to be taken into consideration in the optimization are only the ones already stored in the caching storage and the ones currently being requested by users. Let $\mathcal{F}$ denote the set of these files. Then, each F-AP can simply implement the iterative algorithm with regard to the files in $\mathcal{F}$.

By taking the finite time horizon T as an optimization period, the proposed distributed edge caching scheme in continuous optimization periods is presented in Algorithm 1. In the beginning of each optimization period, each F-AP firstly reads the user request status of the last optimization period. Assume that the change of content popularity is relatively slow compared with the time scale of the optimization period. The corresponding caching policy can then be obtained according the statistics of the last optimization period. After that, each F-AP obtains the file set $\mathcal{F}$. For each file in the set, the iterative algorithm is conducted, and the corresponding caching control is obtained. Finally, the final optimal caching policies subject

Algorithm 1 The proposed distributed edge caching scheme

```
Input: Initial state $\boldsymbol{x}(0)$, iteration termination threshold $\varepsilon$, caching capacity $C$;
Output: $\boldsymbol{c}^*(t)$;
for optimization period $p$ = 1 to $P$ do
    Read the recorded request status $\mathrm{req}_k\left(t\right)$;
    Compute the file set $\mathcal{F} = \left\{ n \,\middle|\, s_n\left(t\right) > 0, \text{or} \sum\limits_{t=(p-1)T}^{pT} q_n\left(t\right) > 0 \right\}$ that requires to solve the PDEs;
    for $n \in \mathcal{F}$ do
        $\phi_n^*\left(t, s_n(t)\right), v_n\left(t, \boldsymbol{x}_n(t)\right), t \in \left[pT, \left(p+1\right)T\right) \leftarrow$ MEANFIELD($p, \varepsilon, \boldsymbol{x}_n(pT)$);
    end for
    $\boldsymbol{c}^*(t), t \in [pT, (p+1)T) \leftarrow$ KNAPSACK($p$, $\boldsymbol{v}(t, \boldsymbol{x}(t))$, $\boldsymbol{\phi}^*(t, \boldsymbol{s}(t))$, $\boldsymbol{s}(pT)$, $C$).
end for
```

to the caching capacity constraint are obtained by using the greedy algorithm to solve the fractional knapsack problem in each time slot. Note that the optimization can be executed in a distributed manner without obtaining the specific states of any other individual F-APs.

B. Mean Field Game for Edge Caching

1) Mean Field Game Formulation: A mean field game can be established based on four conditions [27]: (1) rationality of players; (2) continuum of players; (3) exchangeability among states of players; (4) interactions among players of the mean field type. The first condition can be guaranteed since each F-AP makes logical decisions. The second condition can be assured according to the ultra-dense deployment of F-APs. The third condition holds since the contribution of individuals become infinitesimal among the mass of players and the interchange among the indices of F-APs does not alter the outcome of the game. The fourth condition is assured since each F-AP interacts with the mean field instead of actual individuals in the game.

Given the properties presented above, the original SDG can be converted to an MFG, in which each player is indistinguishable among the mass of players. Therefore, we can consider a generic player by dropping the F-AP index. The interactions between the generic player and the others in the MFG are sufficiently characterized by its own state and the statistical distribution of the mass state. Specifically, $\boldsymbol{x}_n = \left(\boldsymbol{g}^{\mathrm{T}}, s_n\right)^{\mathrm{T}} \in \mathcal{X}$ denotes the state of the generic player, and $m_n \in \mathcal{X}_T, \mathcal{X}_T = \mathcal{X} \times [0, T]$ represents the mean field distribution w.r.t. file n. m_n depends upon both time t and state $\boldsymbol{x}_n(t)$, and it can be expressed as follows:

$$m_n(t, \boldsymbol{x}_n(t)) = \lim_{I \to \infty} \frac{1}{I} \sum_{i=1}^{I} \delta_{\boldsymbol{x}_{i,n}(t)}, \tag{17}$$

where $\delta_{\boldsymbol{x}_{i,n}(t)}$ is the Dirac measure. Note that the mean field distribution is the probability distribution of the states of all the players in the game. The evolution of $m_n(t, \boldsymbol{x}_n(t))$ corresponds to the Fokker-Planck-Kolmogorov (FPK) PDE as follows [28]–[30]:

$$\begin{aligned} \partial_t m_n = \frac{1}{2}\sigma_g^2 \partial_{gg}^2 m_n - \frac{1}{2}\varsigma(\mu_g - g\left(t\right))\partial_g m_n \\ - S\left(c_n\left(t\right) - e^{a-1} a^{q_n(t)}\right)\partial_s m_n. \end{aligned} \tag{18}$$

Let $\bar{g}_{-,k}\left(t\right)$, $\bar{s}_{-,n}\left(t\right)$ denote the average channel state and caching state of the other F-APs, respectively. Given the mean field distribution, the average states in the mean field can be approximated as functions of $m_n(t, \boldsymbol{x}_n(t))$ as follows [31]:

$$\left\{ \begin{array}{l} \bar{g}_{-,k}\left(t\right) = \int g_k\left(t\right) m_n(t, \boldsymbol{x}_n(t)) \mathrm{d}g_k, \\ \bar{s}_{-,n}\left(t\right) = \int s_n\left(t\right) m_n(t, \boldsymbol{x}_n(t)) \mathrm{d}s_n. \end{array} \right. \tag{19}$$

With the establishment of the mean field, each player can obtain its caching control based solely on the local information, including its local state and the mean field distribution. Correspondingly, the optimization problem of each player can be reformulated as a generic problem as follows:

$$\min_{c_n(0 \to T)} \mathcal{J}_n\left(c_n, \boldsymbol{x}_n, m_n\right). \tag{20}$$

Note that the cost function of the MFG here is of the same form with the cost function of the SDG in (11) after dropping the F-AP index i, except for the channel and cache state of other F-AP, i.e., $\bar{g}_{-,k}\left(t\right), \bar{s}_{-,n}\left(t\right)$, are replaced with the mean state via (19). Therefore, the corresponding cost function for the mean field function is not specifically provided here. Besides, the objective function in the MFG depends on the local control, local state and mean field distribution instead of the states of the other players.

The solution to the generic problem in (20) can be obtained by solving the following HJB PDE:

$$\partial_t v_n\left(t, \boldsymbol{x}_n(t)\right) + \min_{c_n(t)} \left[L v_n\left(t, \boldsymbol{x}_n(t)\right) + J_n\left(t\right)\right] = 0, \tag{21}$$

where the differential term $L v_n\left(t, \boldsymbol{x}_n(t)\right)$ can be expressed as follows:

$$\begin{aligned} L v_n\left(t, \boldsymbol{x}_n(t)\right) = \frac{1}{2}\sigma_g^2 \partial_{gg}^2 v_n + \frac{1}{2}\varsigma(\mu_g - g\left(t\right))\partial_g v_n \\ + S\left(c_n\left(t\right) - e^{a-1} a^{q_n(t)}\right)\partial_s v_n. \end{aligned} \tag{22}$$

With the value function $v_n\left(t, \boldsymbol{x}_n(t)\right)$ obtained by solving the HJB PDE in (21), the optimal caching control can be obtained according to the following theorem.

Theorem 1: For the formulated MFG, the optimal caching control can be expressed as follows:

$$c_n^*\left(t\right) = \phi_n^*\left(t, s_n(t)\right) = -\frac{1}{\eta_2}\left(\frac{S}{\omega_2}\partial_s v_n + \eta_1\right). \tag{23}$$

Proof: Define the Hamiltonian corresponding to the optimization problem as $G\left(t, c_n, \boldsymbol{x}_n\right) = L v_n\left(t, \ \boldsymbol{x}_n(t)\right) + J_n\left(t\right)$. As can be observed from (21), the optimal caching control $c_n^*\left(t\right)$ can be obtained by minimizing the Hamiltonian. Compute the derivative of the Hamiltonian with respect to $c_n(t)$. Then, it can be concretely expressed as follows:

$$\frac{\partial G\left(t, c_n, \boldsymbol{x}_n\right)}{\partial c_n} = \omega_2\left(\eta_1 + \eta_2 c_n\left(t\right)\right) + S\partial_s v_n. \tag{24}$$

It can be easy to verify that the Hamiltonian is concave with respect to $c_n(t)$. Let the derivative to be zero. Then, the optimal caching control can be concretely derived as shown in (23). This completes the proof. ∎

By substituting $c_n^*(t)$ in (23) back into (21), the final form of the HJB without variable $c_n(t)$ can be expressed in (25) as shown at the bottom of this page, where

$$\begin{aligned}\hat{O}_n = &\left(q_{i,n}(t)\left(C_1(t, s_{i,n}(t)) + C_3(t, s_{i,n}(t), s_{-i,n}(t))\right)\right. \\ &\left. + q_{i,n}^-(t) C_4(t, s_{i,n}(t), s_{-i,n}(t))\right) \times (S - s_{i,n}(t)). \quad (26)\end{aligned}$$

As can be seen from (18), (23), (21) and (19), the solution of the HJB PDE affects the FPK PDE through the caching control $c_n(t)$, and the solution of the FPK PDE determines the HJB PDE through the total cost function $J_n(t)$. The two coupled PDEs comprehensively form the structure of the mean field: the backward HJB models the induction process of the optimization of each individual, while the forward FPK models the evolution of the mean field as a whole. Therefore, the caching control in one optimization period T can be obtained by iteratively computing the HJB-FPK PDEs. The HJB and FPK equations are regular parabolic partial differential equations that can be solved with the well-established algorithms. We have addressed this issue in the part of simulation results that the PDEPE toolbox in MATLAB (which uses the Method of Lines for PDEs) has been utilized to solve the HJB and the FPK.

The solution of the formulated MFG is the NE that is defined below.

Definition 2 (NE of the MFG): In the MFG, the set of caching control $c_n^*(t) = \phi_n^*(t, s_n(t))$ constitutes the NE of the MFG, i.e., the mean field equilibrium, and $\{\boldsymbol{x}_n^*(t), t \in [0, T]\}$ is the corresponding state trajectory if the following inequality holds:

$$\mathcal{J}(c_n{}^*, \boldsymbol{x}_n^*, m_n) \le \mathcal{J}(c_n, \boldsymbol{x}_n^*, m_n). \quad (27)$$

When the number of players I is sufficiently large, the NE of the MFG is equivalent of the NE of the SDG [28]. Furthermore, the solution pair of the HJB-FPK PDEs in (21) and (18), i.e., $(v_n(t, \boldsymbol{x}_n(t)), m_n(t, \boldsymbol{x}_n(t)))$, can represent the NE of the MFG, exchangeably with $c_n^*(t) = \phi_n^*(t, s_n(t))$.

The iterative algorithm for obtaining the mean field equilibrium is presented in Algorithm 2.

Algorithm 2 Iterative algorithm for the mean field equilibrium

```
function MEANFIELD($p$, $\varepsilon$, $\boldsymbol{x}_n(pT)$)
    Compute $m_n(pT, \boldsymbol{x}_n(pT))$ through (17);
    Set iteration number $q = 1$, $v_n((p+1)T, \cdot) = 0$;
    while $\left|v_n^q(t, \boldsymbol{x}_n(t)) - v_n^{q-1}(t, \boldsymbol{x}_n(t))\right| > \varepsilon$ and $\left|m_n^q(t, \boldsymbol{x}_n(t)) - m_n^{q-1}(t, \boldsymbol{x}_n(t))\right| > \varepsilon$ do
        Solve the HJB in (21), obtain $v_n^q(t, \boldsymbol{x}_n(t))$;
        Update the caching control $\phi_n^q(t, s_n(t))$ through (23);
        Solve the FPK in (18), obtain $m_n^q(t, \boldsymbol{x}_n(t))$;
        Update the mean filed state through (19);
        Update the total cost function $J_n(t)$ through (11), (7), (9);
        $q = q + 1$;
    end while
    Output: $\phi_n^*(t, s_n(t)), v_n(t, \boldsymbol{x}_n(t)), \quad t \in [pT, (p+1)T)$ .
end function
```

2) Mean Field Equilibrium: The NE of the MFG is analysed by firstly discussing the solutions of the HJB and FPK separately, and secondly discussing the existence and uniqueness of the solution pair $(v_n(t, \boldsymbol{x}_n(t)), m_n(t, \boldsymbol{x}_n(t)))$ by considering the interaction between the two PDEs.

Theorem 2: In our formulated MFG, the existence and uniqueness of the value function $v_n(t, , \boldsymbol{x}_n(t))$ is guaranteed for the HJB in (21).

Proof: The value function $v_n(t, \boldsymbol{x}_n(t))$ is the unique viscosity solution of the HJB in (21), if the following conditions hold [32]:

(1) The caching control space $\mathcal{C}$ is a compact subset of $\mathbb{R}$;

(2) The drift term of the state dynamics $\boldsymbol{f}$ is bounded and Lipschitz continuous in $\boldsymbol{x}_n \in \mathcal{X}$, i.e., for $\boldsymbol{f} : [0, T] \times \mathcal{X} \times \mathcal{C} \to \mathcal{X}$, it holds that $|\boldsymbol{f}(t, \boldsymbol{x}_n, c_n)| \le A, |\boldsymbol{f}(t, \boldsymbol{x}_n, c_n) - \boldsymbol{f}(t, \hat{\boldsymbol{x}}_n, c_n)| \le A|\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n|$;

(3) The cost function is bounded and Lipschitz continuous in $\boldsymbol{x}_n \in \mathcal{X}$, with a similar definition of Lipschitz continuity with the second condition.

For the first condition, the control space of the game is $\mathcal{C} = [0, 1]$, which is indeed a compact subset of $\mathbb{R}$.

For the second condition, according to (1) and (3), the drift term in the formulated MFG can be expressed as follows:

$$\begin{cases} f_1(t, g_k, c_n) = \frac{1}{2}\varsigma(\mu_g - g_k(t)), \\ f_2(t, s_n, c_n) = S\left(c_n(t) - e^{a-1}a^{q_n(t)}\right), \end{cases} \quad (28)$$

where $g_k(t)$ is bounded and $c_n(t) \in [0, 1]$. It can be observed that f_1 is bounded and $f_2 \in [-S, 0)$. Therefore, $|\boldsymbol{f}(t, \boldsymbol{x}_n, c_n)| = \sqrt{f_1{}^2 + f_2{}^2} \le A$. Moreover, it can be also observed that $|f_1(t, g_k, c_n) - f_1(t, \hat{g}_k, c_n)| = -\frac{1}{2}\varsigma(g_k - \hat{g}_k)$ and $|f_2(t, s_n, c_n) - f_2(t, \hat{s}_n, c_n)| \equiv 0$. Let $\Delta\boldsymbol{f} = \boldsymbol{f}(t, \boldsymbol{x}_n, c_n) - \boldsymbol{f}(t, \hat{\boldsymbol{x}}_n, c_n)$. Then, we have:

$$|\Delta\boldsymbol{f}(t, \boldsymbol{x}_n, c_n)| = \sqrt{|\Delta f_1(t, g_k, c_n)|^2 + |\Delta f_2(t, s_n, c_n)|^2} \quad (29)$$

$$\begin{aligned}&\partial_t v_n + \frac{1}{2}\sigma_g{}^2\partial_{gg}^2 v_n + \frac{1}{2}\varsigma(\mu_g - g(t))S\partial_g v_n + S\left(-\frac{1}{\eta_2}\left(\frac{S}{\omega_2}\partial_s v_n + \eta_1\right) - e^{a-1}a^{q_n(t)}\right)\partial_s v_n \\ &\qquad + \eta_1\left(-\frac{1}{\eta_2}(S\partial_s v_n + \eta_1)\right) + \frac{\eta_2}{2}\left(-\frac{1}{\eta_2}(S\partial_s v_n + \eta_1)\right)^2 + \hat{O}_n + D_n = 0, \quad (25)\end{aligned}$$

$$= \left|\frac{1}{2}\varsigma\left(g_k - \hat{g}_k\right)\right| \tag{30}$$

$$\le \left|\frac{1}{2}\varsigma\right| \sqrt{\left|g_k - \hat{g}_k\right|^2 + \left|s_n - \hat{s}_n\right|^2}. \tag{31}$$

Correspondingly, the second condition holds for the HJB.

For the third condition, it requires that the map: $J_n : [0,T] \times \mathcal{X} \times \mathcal{C} \to \mathbb{R}$, is bounded and Lipschitz continuous. Firstly, according to (7), (9) and (11), it can be observed that $J_n(t)$ is an elementary function of $\boldsymbol{x}_n(t)$. Therefore, it is bounded. Secondly, the partial derivatives of $J_n(t)$ can be expressed as follows:

$$\begin{aligned}
&\partial_{\boldsymbol{x}} J_n\left(t\right) = \omega_1 \partial_{\boldsymbol{x}} D_n\left(t\right) + \omega_2 \partial_{\boldsymbol{x}} O_n\left(t\right) \\
&= \left[\omega_1 \partial_g D_n\left(t\right) + \omega_2 \partial_g O_n\left(t\right), \omega_1 \partial_s D_n\left(t\right) + \omega_2 \partial_s O_n\left(t\right)\right]^{\mathrm{T}}.
\end{aligned} \tag{32}$$

In (32), $\partial_s D_n(t)$ and $\partial_s O_n(t)$, i.e., the derivatives of $D_n\left(t\right)$ and $O_n\left(t\right)$ w.r.t. $s_n(t)$, can be expressed in (33) and (34) as shown at the bottom of this page, where

$$\partial_s C_1\left(t, s_n\left(t\right)\right) = \tilde{H}^{'}\left(s_n\left(t\right) - \frac{1}{2}S\right), \tag{35}$$

$$\begin{aligned}
\partial_s C_2\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right) = -\tilde{H}^{'}\left(\frac{1}{2}S - s_n\left(t\right)\right) \\
\times \left(1 + e^{-2k\left(s_{-,n}(t) - \frac{1}{2}S\right)}\right)^{-1},
\end{aligned} \tag{36}$$

$$\begin{aligned}
\partial_s C_3\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right) = -\tilde{H}^{'}\left(\frac{1}{2}S - s_n\left(t\right)\right) \\
\times \left(1 + e^{-2k\left(\frac{1}{2}S - s_{-,n}(t)\right)}\right)^{-1},
\end{aligned} \tag{37}$$

$$\begin{aligned}
\partial_s C_4\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right) = \tilde{H}^{'}\left(s_n\left(t\right) - \frac{1}{2}S\right) \\
\times \left(1 + e^{-2k\left(\frac{1}{2}S - s_{-,n}(t)\right)}\right)^{-1},
\end{aligned} \tag{38}$$

$$\tilde{H}^{'}\left(x\right) = \mathrm{d}_x \tilde{H}\left(x\right) = 2ke^{-2kx}\left(1 + e^{-2kx}\right)^{-2}. \tag{39}$$

Since (33) and (34) are both elementary functions of $s_n(t)$, it can be guaranteed that $D_n\left(t\right)$, $O_n\left(t\right)$, $\partial_s D_n\left(t\right)$ and $\partial_s O_n\left(t\right)$ is bounded. Their partial derivatives w.r.t. $g_k(t)$ can be verified to be bounded similarly. Then, the total cost function $J_n(t)$ and its partial derivative w.r.t. the state $\boldsymbol{x}_n(t)$ is bounded. Therefore, $J_n(t)$ is bounded and Lipschitz continuous in $\boldsymbol{x}_n \in \mathcal{X}$.

At this point, all the three conditions have been proven to hold for the HJB in (21). Therefore, the existence and uniqueness can be guaranteed for the solution of the HJB. This completes the proof. ■

Theorem 3: In the formulated MFG, the existence and uniqueness of the mean field distribution $m_n(t,\ \boldsymbol{x}_n(t))$ is guaranteed for the FPK in (18).

Proof: Consider a parabolic partial differential equation of the following form:

$$\begin{aligned}
&m_t + L_m = f \quad \text{in } \mathcal{X}_T, \\
&L_m = -\sum_{i,j=1}^{n} a^{i,j} m_{x_i x_j} + \sum_{i=1}^{n} b^i m_{x_i} + cm,
\end{aligned} \tag{40}$$

where $a^{i,j}, b^i, c \in L^{\infty}\left(\mathcal{X}_T\right), f \in L^2\left(\mathcal{X}_T\right), a^{i,j} = a^{j,i}$. Then, there exists a weak unique solution $m\left(t,x\right)$ [32], [33]. It can be observed from the FPK in (18) that:

$$c = f = 0,\ a^{i,j} = \begin{cases} \frac{1}{2}{\sigma_g}^2, & i = j = 1, \\ 0, & \text{otherwise}. \end{cases} \tag{41}$$

Correspondingly, $\left\|a^{i,j}\right\|_{\infty} \le \frac{1}{2}{\sigma_g}^2 < \infty, \left\|c\right\|_{\infty} = 0 < \infty, \left\|f\right\|_2 = 0 < \infty$. Therefore, the conditions $a^{i,j}, c \in L^{\infty}\left(\mathcal{X}_T\right), f \in L^2\left(\mathcal{X}_T\right), a^{i,j} = a^{j,i}$ are satisfied. For the drift term b^i, its boundedness has been shown in the proof of Theorem 2. Subsequently, given $\left\|b^i\right\|_{\infty} = \sup\left\{b^i\right\}$, the condition $b^i \in L^{\infty}$ can also be satisfied. Therefore, the existence and uniqueness of the solution for the FPK in (18) can be proved. This completes the proof. ■

According to the previous discussions, we have shown that the two PDEs are coupled via the caching control and total cost function, and the optimal caching control can be obtained by iteratively computing the solution pair of the HJB-FPK PDEs, i.e., $\left(v_n\left(t, \boldsymbol{x}_n(t)\right), m_n\left(t, \boldsymbol{x}_n(t)\right)\right)$. By taking the temporary solutions obtained in the last iteration as the initial values of the next iteration, each iteration actually defines a mapping $f : \mathcal{X}_T \times \mathcal{X}_T \to \mathcal{X}_T \times \mathcal{X}_T$. Theorem 2 and Theorem 3 have verified the existence of the unique solutions to the HJB and FPK separately with certain parameters in one iteration, respectively. However, a sequence of solutions may be updated from iterations while solving the coupled PDEs, which may result in divergence of the iteration. In the following, the uniqueness of the solution pair to the two interacting PDEs needs to be discussed.

For a stochastic differential game of large population, e.g., our formulated MFG, the mapping f defined by the HJB-FPK solving process can be proved to be a contraction mapping with the following assumptions [34]:

$$\begin{aligned}
\partial_s D_n\left(t\right) = \sum_{\substack{k \in \mathcal{U}^i \\ \mathrm{req}_k(t) = n}} &\left\{\left[\partial_s C_1\left(t, s_n\left(t\right)\right) + \partial_s C_3\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right)\right]\left[\frac{S}{R_k\left(t\right)} + \frac{S - s_n\left(t\right)}{R^F}\right]\right. \\
&\left. - \frac{C_1\left(t, s_n\left(t\right)\right) + C_3\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right)}{R^F} + \partial_s C_2\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right)\left[\frac{S}{R_{-,k}\left(t\right)} + \frac{S - s_{-,n}\left(t\right)}{R^F}\right]\right\},
\end{aligned} \tag{33}$$

$$\begin{aligned}
\partial_s O_n\left(t\right) = &\eta\left(q_n\left(t\right)\left(\partial_s C_1\left(t, s_n\left(t\right)\right) + \partial_s C_3\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right)\right) + q_n^{-}\left(t\right)\partial_s C_4\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right)\right) \\
&\times\left(S - s_n\left(t\right)\right) - \eta\left(q_n\left(t\right)\left(C_1\left(t, s_n\left(t\right)\right) + C_3\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right)\right) + q_n^{-}\left(t\right)C_4\left(t, s_n\left(t\right), s_{-,n}\left(t\right)\right)\right),
\end{aligned} \tag{34}$$

(1) The caching control space $\mathcal{C}$ is a compact interval;
(2) The drift term of the state dynamics and the total cost function are bounded and Lipschitz continuous in $\boldsymbol{x}_n \in \mathcal{X}$;
(3) The first and second order derivatives of the drift term of the state dynamics and the cost function are bounded and Lipschitz continuous in $\boldsymbol{x}_n \in \mathcal{X}$;
(4) The drift term of the state dynamics is Lipschitz continuous in $c_n \in \mathcal{C}$;
(5) For any $q \in \mathbb{R}$ and any probability distribution $\rho(\mathrm{d}\boldsymbol{x}_{-,n}) \in \mathbb{R}$, the set $\mathcal{S} = \arg\min_{c_n \in \mathcal{C}} \left\{ \int_{\boldsymbol{x}_{-,n}} [qf + J] \rho(\mathrm{d}\boldsymbol{x}_{-,n}) \right\}$ is a singleton, and the resulting c_n, as a function on $(\boldsymbol{x}_n, q)$, is Lipschitz continuous in $(\boldsymbol{x}_n, q) \in \mathcal{X} \times \mathbb{R}$.

Given the above assumptions, we have the following theorem.

Theorem 4: The existence and uniqueness of the NE are guaranteed in the formulated MFG.

Proof: The assumptions (1)-(5) can be satisfied in our formulated MFG with a detailed proof similar to Theorem 2. Then, according to the fixed-point theorem, given this contraction mapping, there exists a unique fixed-point, which can be found by starting from an arbitrary point $(v_n^0(t, \boldsymbol{x}_n(t)), m_n^0(t, \boldsymbol{x}_n(t)))$. Therefore, the iterations to solve (21) and (18) can be guaranteed to converge to a fixed-point $(v_n^*(t, \boldsymbol{x}_n(t)), m_n^*(t, \boldsymbol{x}_n(t)))$, which is the unique solution pair of the HJB-FPK PDEs. Correspondingly, the existence and uniqueness of the NE can be guaranteed in the formulated MFG. This completes the proof. ■

C. Fractional Knapsack Problem for Edge Caching

At this point, the optimal caching control $c_n^*(t) = \phi_n^*(t, s_n(t))$ for each file can be obtained as a mapping from time slot and caching state to caching policy, without considering the caching capacity constraint. By taking into account the caching capacity C, the optimization of caching policies for all the files in each time slot can be considered to be a knapsack problem, in which the caching decision of a file fraction depends on its weight and value. Therefore, the final optimal caching policies for all the files in an optimization period T can be obtained by solving a sequence of knapsack problems based on the solutions of the MFG in Section IV-B.

The formulated knapsack problem in each time slot can be expressed as follows:

$$\max_{\boldsymbol{c}(t)} \sum_{n \in \mathcal{N}} c_n(t) \nu_n(t) \tag{42}$$

$$\text{s.t.} \sum_{n \in \mathcal{N}} c_n(t) \varpi_n(t) \le C, \tag{42a}$$

$$c_n(t) \in [0, 1], \tag{42b}$$

where each F-AP decides the caching policy of each file to maximize the overall value under the caching capacity constraint (42a) and the caching policy value constraint (42b). Let $\varpi_n(t)$ and $\nu_n(t)$ denote the weight and value of file n, respectively, and they can be expressed by the caching control and the value function obtained in Algorithm 2, respectively. Specifically, the weight is the new fraction of file n to be cached, i.e., the value of the caching control that corresponds to the current time slot t and caching state $s_n(t)$. The value $\nu_n(t)$ is the cost reduction induced by caching file n. Let $s_n^-(t+1)$ and $s_n^+(t+1)$ denote the resulting caching state in the next time slot if file n is chosen to and not to cache file n, respectively. Then, the value can be expressed as $\nu_n(t) = v_n(t+1, s_n^-(t+1)) - v_n(t+1, s_n^+(t+1))$. Note that $\phi_n^*(t, s_n(t))$ and $v_n(t, s_n(t))$ can be obtained from Algorithm 2, while $s_n^-(t+1)$ and $s_n^+(t+1)$ can be obtained based on the caching control value $\phi_n^*(t, s_n(t))$ and the caching state $s_n(t)$ of the current time slot via (3).

Considering that the caching state and policy of each file take continuous values, we can solve the corresponding fractional knapsack problem by using a greedy algorithm, in which the files with larger values per unit weight will be cached with higher priority. Let r denote the current remaining storage of the considered F-AP. Firstly, the files are sorted in a descending order by the value per unit weight. Then, for each file, the optimal caching policy $c_n^*(t)$ is computed while the remaining storage r is updated according to the following rules:

(1) If the remaining storage r is larger than $\varpi_n(t)$, then the optimal caching policy $c_n^*(t)$ is set to $\varpi_n(t)$.
(2) Else if the remaining storage r is larger than 0, then the optimal caching policy $c_n^*(t)$ is set to r.
(3) Otherwise, the optimal caching policy $c_n^*(t)$ is set to 0.

After solving the fractional knapsack problem in the current time slot, the caching state of the next time slot can be computed according to (3). Therefore, by successively solving the fractional knapsack problems in each time slot, the final optimal caching policy in the optimization period and the corresponding state trajectory can be obtained. The proposed greedy algorithm is presented in Algorithm 3.

D. Optimality and Complexity Analysis

Firstly, the optimality of the proposed distributed edge caching scheme as presented in Algorithm 1 is analysed.

Theorem 5: The final caching policy obtained in Algorithm 1 is optimal.[3]

Proof: The proposed caching scheme in Algorithm 1 is composed of two steps: firstly, obtain the optimal caching control for each file without considering the caching capacity by using Algorithm 2; secondly, obtain the optimal caching policy for each file taking the caching capacity into consideration by using Algorithm 3. Therefore, we need to prove the optimality of Algorithm 2 and Algorithm 3, respectively, and then the optimality of Algorithm 1.

1) Firstly, as shown in Section IV-B, we have proven the existence and uniqueness of the mean field equilibrium. Due to the equivalence of MFG NE and SDG NE, the optimal solution can be obtained for the SDG in (12). Therefore, the optimality of the caching control obtained by using Algorithm 2 can be guaranteed.

[3] The optimality here means that Algorithm 1 can obtain the optimal solution for (12) and satisfy the caching capacity constraint.

Algorithm 3 Greedy algorithm for the knapsack problem

```
function KNAPSACK($p$, $\boldsymbol{v}(t, \boldsymbol{s}(t))$, $\boldsymbol{\phi}^*(t, \boldsymbol{s}(t))$, $\boldsymbol{s}(pT)$, $C$)
    for $t \in [pT, (p+1)T)$ do
        $\boldsymbol{\varpi} = \boldsymbol{\phi}(t, \boldsymbol{s}(t))$;
        $\boldsymbol{s}^-(t+1) = \boldsymbol{s}(t) - Se^{a-1}a^{\boldsymbol{q}(t)}\Delta t$;
        $\boldsymbol{s}^+(t+1) = \boldsymbol{s}(t) + S\boldsymbol{\phi}^*(t, \boldsymbol{s}(t))\Delta t$;
        $\boldsymbol{\nu} = \boldsymbol{v}(t+1, \boldsymbol{s}^-(t+1)) - \boldsymbol{v}(t+1, \boldsymbol{s}^+(t+1))$;
        $r = C - \sum_n s_n(t)$;
        Sort the files in a descending order by $\nu_n/\varpi_n$;
        for $n$ = 1 to $N$ do
            if $r > \varpi_n$ then
                $c_n^*(t) = \varpi_n$;
            else if $r > 0$ then
                $c_n^*(t) = r$;
            else
                $c_n^*(t) = 0$;
            end if
            $r = r - c_n^*(t)$;
        end for
        $\boldsymbol{s}(t+1) = \boldsymbol{s}(t) + S\left(\boldsymbol{c}^*(t) - e^{a-1}a^{\boldsymbol{q}(t)}\right)\Delta t$.
    end for
    Output: $\boldsymbol{c}^*(t), t \in [pT, (p+1)T)$.
end function
```

2) Secondly, according to the knapsack theory, the greedy algorithm can work out the optimal solution of a fractional knapsack problem according to [35]. Therefore, the optimality of Algorithm 3 can be guaranteed.
3) Finally, we need to prove the optimality by combing Algorithm 2 and Algorithm 3. In Algorithm 2, the obtained optimal caching control is actually a mapping, i.e., $\phi_n^*(t, s_n(t))$, from the pair of time slot t and caching state $s_n(t)$ to the caching policy value. In Algorithm 3, the final caching policy that takes the caching capacity constraint of each F-AP into account is deduced together with the caching state trajectory of each file according to the mapping $\phi_n^*(t, s_n(t))$. Therefore, the mean field equilibrium is not violated.

In summary, the mapping of the optimal caching control that solves the problem in (12) without considering the caching capacity constraint is calculated in Algorithm 2, while the actual optimal caching policy considering the caching capacity constraint is calculated in Algorithm 3 based on the obtained mapping in Algorithm 2. This completes the proof. ■

Secondly, the computational complexity is analysed by considering the iterative algorithms for the mean field equilibrium and the greedy algorithm for the knapsack problem. Note that although the set of files that needs PDE solving is reduced to $\mathcal{F}$ as stated in Section IV-A, it is difficult to obtain the analytical expression for the number of files in $\mathcal{F}$. Therefore, for the iterative algorithm in Algorithm 2, the worst complexity is still linear with the size of content library, which is of $O(N)$. For the greedy algorithm in Algorithm 3, the complexity is of $O(N \log N)$ due to the sorting process. Therefore, the overall complexity of the proposed scheme, i.e., Algorithm 1, is of $O(N \log N)$, which is lower compared with the complexity $O(NI)$ of the schemes that require information exchange among ultra-densely deployed F-APs.

V. Simulation Results

The proposed distributed edge caching scheme is verified through simulations, in which a circular area with the radius of 600 m is considered. For the sake of simplicity, we assume that all the F-APs have the same transmission power P [4]. Unless otherwise stated, the system parameters are set as follows: $P = 1$ W, $\sigma^2 = -77$ dBm, $W = 10$ MHz, $K = 100$, $S = 100$ MB, $\lambda = 0.1$, $a = 0.1$, $\eta_1 = 2.5 \times 10^3, \eta_2 = 1.3 \times 10^8$, $\omega_1 = 100$, $\omega_2 = 10^{-6}$, $\varepsilon = 0.005$. Moreover, the coupled HJB-FPK PDEs are iteratively solved to the fixed-point by utilizing the PDEPE toolbox of MATLAB. Due to the close distance between users and F-APs in ultra-dense F-RANs, a static channel model is assumed in the simulations, thus the dynamic states of the F-APs can be simply characterized by the caching states [36], [37]. Assume that the probability of a user requesting file n in one optimization period follows Zipf distribution, i.e., $p_n = \frac{n^{-\beta}}{\sum_{n=1}^{N} n^{-\beta}}$, where the parameter β depicts the steepness of the distribution with a positive value and a larger β implicates a more skewed distribution of content popularity.

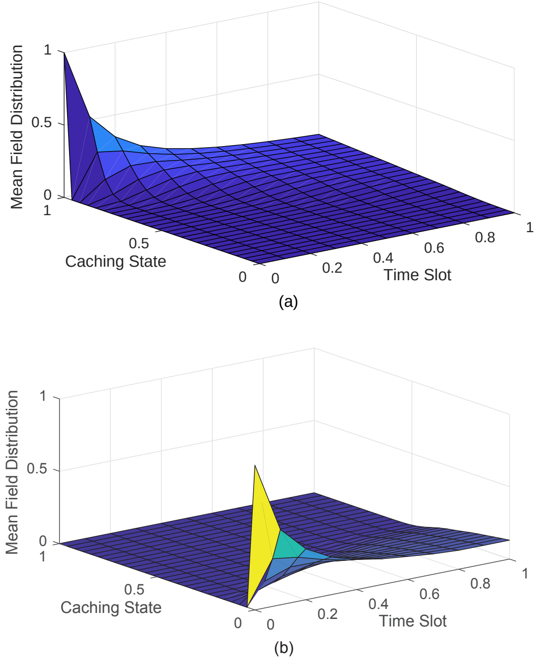
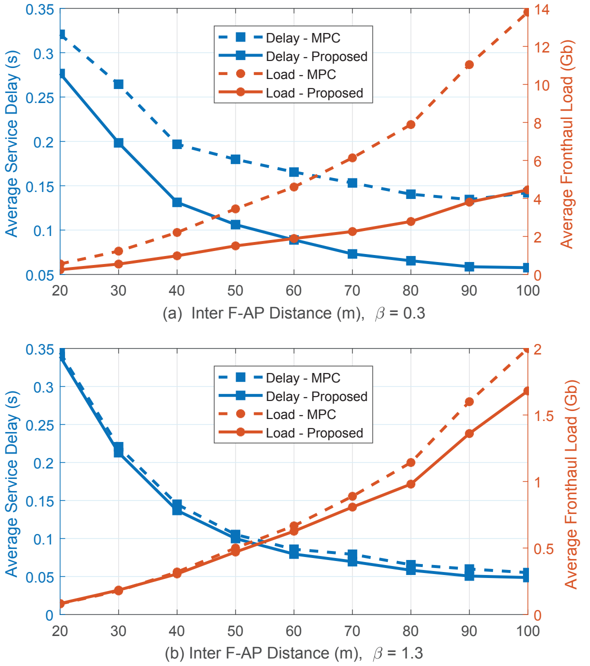


Fig. 2. Mean field distribution at the equilibrium.

In Fig. 2, we show the mean field distribution at the equilibrium with time slot t and normalized caching state s of a certain file under two different popularity and initial caching state conditions. Fig. 2(a) corresponds to the circumstance where the file is comparatively unpopular and is initially stored in the caching storage. It can be observed that the probability of $s = 1$ decreases with time, indicating that F-APs tend not to cache the file due to few requests. In comparison, Fig. 2(b) shows the evolution of the mean field when the file is popular but not cached previously. It can be observed that the probability shifts from $s = 0$ towards a larger s, indicating



(a) Inter F-AP Distance (m), $\beta = 0.3$

(b) Inter F-AP Distance (m), $\beta = 1.3$

Fig. 3. Average request service delay and fronthaul traffic load versus inter F-AP distance for the proposed and MPC schemes under static user requests with $\beta = 0.3$ and $\beta = 1.3$.

that the mass of F-APs has a tendency to cache the file due to its high popularity. These two observations generally validate the ability of F-APs to dynamically update the caching state according to real-time requests.

In Fig. 3, we show the average request service delay and fronthaul traffic load versus average inter F-AP distance (IFD) for the proposed scheme and the most popular caching (MPC) scheme under static user requests with $\beta = 0.3$ and $\beta = 1.3$, respectively. As shown, the average request service delay decreases with IFD, while the average fronthaul traffic load increases with IFD. The reason is that the F-RAN becomes sparse and the inter F-AP interference decreases as IFD increases, whereas the effect of the channel condition degradation is comparatively small because of the generally ultra-dense deployment, which result in increased wireless transmission rate and associated users of each F-AP. Moreover, Fig. 3(a) shows an obvious performance gap between the proposed scheme and the MPC scheme whereas the performance gap is relatively small in Fig. 3(b). The reason is that the MPC scheme performs well only when the user preferences are centralized to a small set of files, thus can be better matched with the Zipf distribution. On the contrary, the proposed scheme shows satisfactory performance with different β, because it is able to capture the real-time request status even when the users show less obvious preference for certain files.

In Fig. 4, we show the average total cost of the proposed and MPC schemes versus caching capacity under static user requests with different skewness. Note that the caching capacity is normalized w.r.t. the size of content library. It can be observed that the average total cost of both schemes decreases with the caching capacity due to the larger amount of possible cached files. However, the proposed scheme exhibits a much slower decrease when the caching capacity becomes larger, whereas the MPC scheme shows a relatively constant cost reduction speed with the increase of caching capacity. Moreover, when $\beta = 0.3$, i.e., the request probability follows a more uniform distribution, the proposed scheme achieves lower cost than the MPC scheme. However, the performance gain over the MPC scheme becomes smaller with a larger capacity. When $\beta = 1.3$, i.e., the request probability follows a more skewed distribution, the performance of the MPC scheme even exceeds the proposed scheme with a larger capacity. This is due to the fact that the proposed scheme computes the optimal caching policy of each file to minimize the cost, hence the obtained caching policy value may be comparatively small to reduce the cost of the fronthaul load. Therefore, when the caching capacity becomes considerably large, it is no longer the main constraint of the performance of the proposed scheme, which leads to the flattened performance. However, the MPC scheme only caches certain files at the beginning of the optimization period, and the load induced by caching is comparatively small. When the caching capacity is large and approaches 1, the MPC scheme almost caches all the files in the library, hence the cache hit rate is considerably large, resulting in low cost, especially under more skewed request probability. Generally, the proposed scheme is more suitable with a relatively small caching capacity under less skewed user requests.

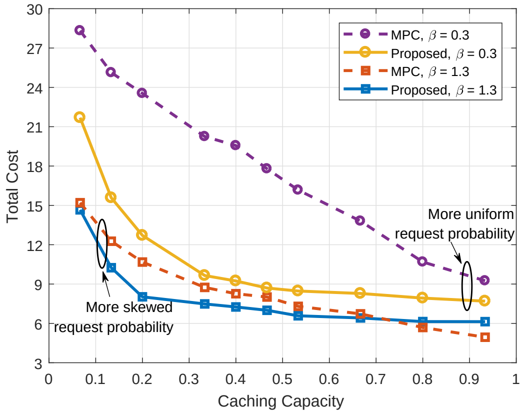


Fig. 4. Average total cost versus caching capacity for the proposed and MPC schemes under static user requests.

In Fig. 5, we show the average total cost of the proposed and four baseline schemes under time-variant user requests, where the user request probability varies in consecutive optimization periods. We set the Zipf distribution parameter $\beta = 1.3$, and choose the MPC, least recently used (LRU), least frequently used (LFU), and first-in-first-out (FIFO) [38] schemes as the four baseline schemes. It can be observed that the proposed scheme shows a stable performance gain over the baseline schemes. Specifically, the average total cost is reduced by 30% to 35% compared with these baseline schemes. The reason is that the MPC scheme employs static caching policy without considering time-variant user requests, and the LRU, LFU and FIFO schemes employ dynamic caching policies only based on local caching state. On the contrary, the proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2961081, IEEE Transactions on Communications

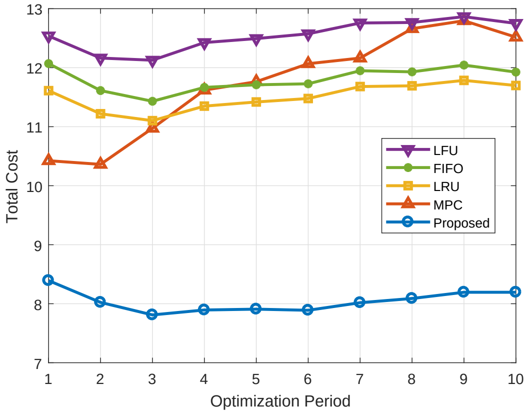


Fig. 5. Average total cost versus optimization period for the proposed and baseline schemes under time-variant user requests.

scheme optimizes the caching policy according to real-time requests and takes into account both local state and mean field distribution. Therefore, the proposed scheme can adapt to time-variant user requests to minimize the average total cost.

VI. Conclusions

In this paper, we have proposed a dynamic distributed edge caching scheme in ultra-dense F-RANs by considering time-variant user requests. Exploiting the ultra-dense property of the F-AP deployment, we have formulated the edge caching optimization problem as an MFG and proposed an iterative algorithm that enables each F-AP to determine its dynamic caching control based on local information without extra communications with adjacent F-APs. Moreover, taking into consideration the caching capacity of F-APs, we have formulated fractional knapsack problems and proposed a greedy algorithm for each F-AP to obtain the final optimal caching policy to minimize both the request service delay and fronthaul traffic load in a distributive manner. Simulation results have shown that our proposed scheme provides considerable performance improvement over the baseline schemes.

References

[1] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. T. Sukhavasi, C. Patel, and S. Geirhofer, "Network densification: The dominant theme for wireless evolution into 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 82–89, Feb. 2014.

[2] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netwk.*, vol. 30, no. 4, pp. 46–53, Jul. 2016.

[3] Y. Ku, D. Lin, C. F. Lee, P. J. Hsieh, H. Y. Wei, C. T. Chou, and A. C. Pang, "5G radio access network design with the fog paradigm: Confluence of communications and computing," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 46–52, Apr. 2017.

[4] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, and L. Hanzo, "Distributed caching for data dissemination in the downlink of heterogeneous networks," *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3553–3568, Oct. 2015.

[5] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Cache placement in Fog-RANs: From centralized to distributed algorithms," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7039–7051, Nov. 2017.

[6] A. Abboud, E. Baştuğ, K. Hamidouche, and M. Debbah, "Distributed caching in 5G networks: An alternating direction method of multipliers approach," in *Proc. IEEE 16th Int. Workshop Signal Processing Advances Wireless Commun. (SPAWC)*, June 2015, pp. 171–175.

[7] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "Joint resource allocation for software-defined networking, caching, and computing," *IEEE/ACM Trans. Netwk.*, vol. 26, no. 1, pp. 274–287, Feb. 2018.

[8] K. Hamidouche, W. Saad, M. Debbah, and H. V. Poor, "Mean-field games for distributed caching in ultra-dense small cell networks," in *Proc. American Control Conf.*, July 2016, pp. 4699–4704.

[9] K. Poularakis and L. Tassiulas, "Code, cache and deliver on the move: A novel caching paradigm in hyper-dense small-cell networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 675–687, Mar. 2017.

[10] W. Wang, R. Lan, J. Gu, A. Huang, H. Shan, and Z. Zhang, "Edge caching at base stations with device-to-device offloading," *IEEE Access*, vol. 5, pp. 6399–6410, Mar. 2017.

[11] Y. Jiang, W. Huang, M. Bennis, and F. Zheng, "Decentralized asynchronous coded caching design and performance analysis in fog radio access networks," *IEEE Trans. Mobile Computing (Early Access)*, pp. 1–11, Jan. 2019.

[12] H. Kim, J. Park, M. Bennis, S. L. Kim, and M. Debbah, "Ultra-dense edge caching under spatio-temporal demand and network dynamics," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[13] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 180–190, Feb. 2018.

[14] J. Li, T. Khoa Phan, W. Koong Chai, D. Tuncer, G. Pavlou, D. Griffin, and M. Rio, "DR-Cache: Distributed resilient caching with latency guarantees," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 441–449.

[15] Y. Huang, X. Song, F. Ye, Y. Yang, and X. Li, "Fair and efficient caching algorithms and strategies for peer data sharing in pervasive edge computing environments," *IEEE Trans. Mobile Comput. (Early Access)*, pp. 1–13, Jan. 2019.

[16] Y. Wang, W. Wang, Y. Cui, K. G. Shin, and Z. Zhang, "Distributed packet forwarding and caching based on stochastic network utility maximization," *IEEE/ACM Trans. Netwk.*, vol. 26, no. 3, pp. 1264–1277, June 2018.

[17] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[18] E. Baştuğ, M. Bennis, and M. Debbah, "Proactive caching in 5G small cell networks," in *Towards 5G: Applications, Requirements and Candidate Technologies*. Wiley, 2016, pp. 78–98.

[19] H. Tembine and M. Debbah, "Reacting to interference: A mean field stochastic game approach," in *Mechanisms and Games for Dynamic Spectrum Allocation*. Cambridge University Press, Dec. 2013, pp. 57–73.

[20] M. de Mari, E. Calvanese Strinati, M. Debbah, and T. Q. S. Quek, "Joint stochastic geometry and mean field game optimization for energy-efficient proactive scheduling in ultra dense networks," *IEEE Trans. Cognitive Commun. Netwk.*, vol. 3, no. 4, pp. 766–781, Dec 2017.

[21] C. Yang, H. Dai, J. Li, Y. Zhang, and Z. Han, "Distributed interference-aware power control in ultra-dense small cell networks: A robust mean field game," *IEEE Access*, vol. 6, pp. 12 608–12 619, 2018.

[22] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "Cache-aware user association in backhaul-constrained small cell networks," in *Proc. Int. Symp. Modeling Opt. Mobile, Ad Hoc Wireless Netwk. (WiOpt)*, May 2014, pp. 37–42.

[23] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1268–1283, Feb. 2019.

[24] P. Mannucci, "Nonzero-sum stochastic differential games with discontinuous feedback," *SIAM J. Control Optim.*, vol. 43, no. 4, pp. 1222–1233, 2004.

[25] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.

[26] D. W. K. Yeung and L. Petrosjan, *Cooperative Stochastic Differential Games*. Springer, 2006.

[27] O. Guéant, "A reference case for mean field games models," *J. De Math. Pures et Appliquées*, vol. 92, no. 3, pp. 276–294, Sept. 2009.

[28] O. Guéant, J.-M. Lasry, and P.-L. Lions, "Mean field games and applications," in *Paris-Princeton Lectures on Mathematical Finance*. Springer, 2011, pp. 205–266.

[29] H. Kim, J. Park, M. Bennis, and S. Kim, "Massive UAV-to-ground communication and its stable movement control: A mean-field approach," in *Proc. IEEE 19th Int. Workshop Signal Processing Advances Wireless Commun. (SPAWC)*, 2018, pp. 1–5.

[30] H. Shiri, J. Park, and M. Bennis, "Massive autonomous UAV path planning: A neural network based mean-field game theoretic approach," *arXiv preprint arXiv:1905.04152*, 2019.

[31] Y. Hu, Y. Jiang, M. Bennis, and F. Zheng, "Distributed edge caching in ultra-dense fog radio access networks: A mean field approach," in *Proc. IEEE VTC 2018 Fall*, Aug. 2018, pp. 1–6.

[32] L. Evans, *Partial Differential Equations*. American Mathematical Society, 1998.

[33] C. Le Bris and P.-L. Lions, "Renormalized solutions of some transport equations with partially w1,1 velocities and applications," *Annali di Matematica Pura ed Applicata*, vol. 183, no. 1, pp. 97–130, Mar. 2004.

[34] M. Huang, R. P. Malhamé, and P. E. Caines, "Large population stochastic dynamic games: Closed-loop mckean-vlasov systems and the nash certainty equivalence principle," *Commun. Inf. Syst.*, vol. 6, no. 3, pp. 221–252, 2006.

[35] B. Korte and J. Vygen, *Combinatorial optimization*. Springer, 2012, vol. 2.

[36] S. Samarakoon, M. Bennis, W. Saad, M. Debbah, and M. Latva-aho, "Energy-efficient resource management in ultra dense small cell networks: A mean-field approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec 2015, pp. 1–6.

[37] ——, "Ultra dense small cell networks: Turning density into energy efficiency," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1267–1280, May 2016.

[38] C. Aggarwal, J. L. Wolf, and P. S. Yu, "Caching on the world wide web," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 1, pp. 94–107, Jan. 1999.

Yanxiang Jiang (S'03-M'07-SM'18) received the B.S. degree in electrical engineering from Nanjing University, Nanjing, China, in 1999 and the M.S. and Ph.D. degrees in communications and information systems from Southeast University, Nanjing, China, in 2003 and 2007, respectively.

Dr. Jiang was a Visiting Scholar with the Signals and Information Group, Department of Electrical and Computer Engineering, University of Maryland at College Park, College Park, MD, USA, in 2014. He is currently an Associate Professor with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China. His research interests are in the area of broadband wireless mobile communications, covering topics such as edge caching, radio resource allocation and management, fog radio access networks, small cells and heterogeneous networks, cooperative communications, green communications, device to device communications, massive MIMO, and machine learning for wireless communications.

Yabai Hu received the M.S. degree in communications and information systems from Southeast University, Nanjing, China.

Her research interests include radio resource management and edge caching.

Mehdi Bennis (S'07-AM'08-SM'15) received his M.Sc. degree in electrical engineering jointly from EPFL, Switzerland, and the Eurecom Institute, France, in 2002. He obtained his Ph.D. from the University of Oulu in December 2009 on spectrum sharing for future mobile cellular systems. Currently he is an associate professor at the University of Oulu and an Academy of Finland research fellow. His main research interests are in radio resource management, heterogeneous networks, game theory, and machine learning in 5G networks and beyond. He has co-authored one book and published more than 200 research papers in international conferences, journals, and book chapters. He was the recipient of the prestigious 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best Paper Award for the Journal of Wireless Communications and Networks, and the 2017 all-University of Oulu Award for Research.

Fu-Chun Zheng (M'95-SM'99) obtained the BEng (1985) and MEng (1988) degrees in radio engineering from Harbin Institute of Technology, China, and the PhD degree in Electrical Engineering from the University of Edinburgh, UK, in 1992.

From 1992 to 1995, he was a post-doctoral research associate with the University of Bradford, UK, Between May 1995 and August 2007, he was with Victoria University, Melbourne, Australia, first as a lecturer and then as an associate professor in mobile communications. He was with the University of Reading, UK, from September 2007 to July 2016 as a Professor (Chair) of Signal Processing. He has also been a distinguished adjunct professor with Southeast University, China, since 2010. Since August 2016, he has been a distinguished professor with Harbin Institute of Technology (Shenzhen), China and the University of York, UK. He has been awarded two UK EPSRC Visiting Fellowships - both hosted by the University of York (UK): first in August 2002 and then again in August 2006. Over the past two decades, Dr Zheng has also carried out many government and industry sponsored research projects - in Australia, the UK, and China. He has been both a short term visiting fellow and a long term visiting research fellow with British Telecom, UK. Dr Zheng's current research interests include signal processing for communications, multiple antenna systems, green communications, and ultra-dense networks.

He has been an active IEEE member since 1995. He was an editor (2001 - 2004) of IEEE Transactions on Wireless Communications. In 2006, Dr Zheng served as the general chair of IEEE VTC 2006-S, Melbourne, Australia (www.ieeevtc.org/vtc2006spring) - the first ever VTC held in the southern hemisphere in VTC's history of six decades. More recently he was the executive TPC Chair for VTC 2016-S, Nanjing, China (the first ever VTC held in mainland China: www.ieeevtc.org/vtc2016spring).

Xiaohu You (SM'11-F'12) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 1982, 1985, and 1988, respectively. Since 1990, he has been working with National Mobile Communications Research Laboratory at Southeast University, where he holds the ranks of professor and director. He is the Chief of the Technical Group of China 3G/B3G Mobile Communication R&D Project. His research interests include mobile communications, adaptive signal processing, and artificial neural networks, with applications to communications and biomedical engineering.

Dr. You was a recipient of the Excellent Paper Prize from the China Institute of Communications in 1987; the Elite Outstanding Young Teacher award from Southeast University in 1990, 1991, and 1993; and the National Technological Invention Award of China in 2011. He was also a recipient of the 1989 Young Teacher Award of Fok Ying Tung Education Foundation, State Education Commission of China.