A Stackelberg Game Approach to Proactive Caching in Large-Scale Mobile Edge Networks

Zijie Zheng, *Student Member, IEEE*, Lingyang Song, *Senior Member, IEEE*, Zhu Han, *Fellow, IEEE*, Geoffrey Ye Li, *Fellow, IEEE*, and H. Vincent Poor, *Fellow, IEEE*

***Abstract*— Caching popular files in the storage of edge networks, namely edge caching, is a promising approach for service providers (SPs) to reduce redundant backhaul transmission to edge nodes (ENs). It is still an open problem to design an efficient incentive mechanism for edge caching in 5G networks with a large number of ENs and mobile users. In this paper, an edge network with one SP, a large number of ENs and mobile users with time-dependent requests is investigated. A convergent and scalable Stackelberg game for edge caching is designed. Specifically, the game is decomposed into two types of sub-games, a storage allocation game (SAG) and a number of user allocation games (UAGs). A Stackelberg game-based alternating direction method of multipliers (Stackelberg game-based ADMM) is proposed to solve either the SAG or each UAG in a distributed manner. Based on both analytical and simulation results, the convergence speed, the optimum of the entire game, and the amount of information exchange are linearly (or sublinearly) related to the network size, which indicates that this framework can potentially cope with large-scale caching problems. The proposed approach also requires less backhaul resource than the existed approaches.**

***Index Terms*— Stackelberg game, alternating direction of multipliers (ADMM), proactive caching, large-scale networks.**

I. Introduction

With recent advances in wireless communication networks, an increasing number of users have been attracted to heterogenous multimedia services through mobile devices. Global mobile data traffic reached 7.2 exabytes (7.2×10^{18} bytes) per month in 2016, of which more than 50% corresponds to mobile multimedia streaming [1], [2]. As a result, limited transmission resources in wireless communication networks cannot support excessive requests for multimedia streaming, especially in the backhaul links among the core networks, i.e., the service providers (SPs) in the Internet, and the edge nodes (ENs), i.e., base stations (BSs), small-cell base stations (SBSs), and WiFi access points (APs).

Fortunately, a significant number of multimedia requests are repetitive among users. The popularity of files usually satisfies Zipf's law [3], under which various users concentrate on a few types of similar files. Thus, the redundancy of file transmission in the backhaul can be greatly reduced when the SPs can generate and cache popular files in edge networks before the files are requested by end users [4]. This brings the files closer to the users. Specifically, during times of light traffic, the SPs can proactively transmit popular files to storage devices at the edge, and those files are delivered from the edge networks to mobile users directly when they are requested. With proper proactive caching strategies, the heavy traffic load can be relieved during times of peak traffic and latency can be reduced as well.

A. Related Work in Edge Caching

A number of techniques have been proposed and applied to edge caching and many ENs in 5G networks have already deployed storage capability [5]–[14]. The work in [6] has proposed the concept of "Femto Caching" and designed a greedy algorithm to achieve a sub-optimal delay reduction through file selection and user access control. In [7] and [8], novel caching approaches exploit physical layer models in the wireless links between the ENs and the users. In addition, device-to-device communications have been further considered in [9] and [10], where the users can obtain their desired files from neighboring users in addition to the ENs. A number of recent works, for example [11], have taken dynamic changes into account. In [12]–[14], edge caching is investigated in the context of stochastic networks via stochastic geometry, where the distributions of the ENs and the users are modeled as Poisson point processes (PPPs). The fluctuations in users' demands have been used in [15] to develop an online caching scheme to update the users' time dependent demands.

In edge caching, the SPs and ENs usually have their own individual benefits when applying caching strategies, and their benefits may conflict with each other, which makes centralized caching approaches difficult to deploy. Caching more files in the storage devices of the ENs can help the

Manuscript received October 27, 2017; revised March 21, 2018 and May 14, 2018; accepted May 16, 2018. Date of publication May 30, 2018; date of current version August 10, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61625101, and in part by the U.S. National Science Foundation under Grants ECCS-1647198, CNS-1443894, CNS-1731017, CNS-1717454, CNS-1731424, CNS-1702850, and CNS-1646607. The associate editor coordinating the review of this paper and approving it for publication was J. Zhang. *(Corresponding author: Zijie Zheng.)*

Z. Zheng and L. Song are with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: zijie.zheng@pku.edu.cn; lingyang.song@pku.edu.cn).

Z. Han is with the Electrical and Computer Engineering Department, University of Houston, Houston, TX 77204 USA, also with the Computer Science Department, University of Houston, Houston, TX 77204 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 02447, South Korea (email: hanzhu22@gmail.com).

G. Y. Li is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: liye@ece.gatech.edu).

H. V. Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

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

Digital Object Identifier 10.1109/TWC.2018.2839111

SPs save more backhaul resources and serve more users directly. But this may jeopardize the benefits to the ENs due to the cost of occupying limited caching storage. Thus, it is necessary to design *incentive mechanisms* whereby the SPs pay for the ENs' storage and backhaul resources. Thus, game theory plays an important role [16]–[20]. Matching-based approaches on file-EN pairing have been developed in [16], where a multiple object auction has been proposed. In [17], a Stackelberg game is considered in which the SPs are the leaders that provide payments to the ENs, the followers, which adjust their storage resources accordingly. A reverse Stackelberg game has been considered in [18], in which the SPs, modeled as the followers, decide the prices based on the storage resources provided by the ENs. In [19], a contract game approach has been used to deal with different types of SPs. The ENs provide various contracts to the SPs based on the popularity of their generated files. In [20], the caching problem has been modeled as a potential game considering the storage and backhaul constraints in each EN. The Nash equilibrium (NE) among various ENs on file caching is achieved with an iterative potential function update in a decentralized way, in which only local information exchange is needed among the ENs.

B. Contributions

The above game theoretic works address the incentive mechanism design problem in edge caching. However, the complexity and scalability of edge caching need to be further considered in future wireless communication networks that will have tremendous numbers of ENs, users, and service demands and involve a huge amount of data. Here, complexity describes the number of iteration steps of an algorithm and the amount of information exchanged among players in the game, while the scalability reflects how the algorithm performs as the network grows.

In this paper, we develop a framework for convergent and scalable incentive mechanism design in edge caching. Specifically, a network with one SP, multiple ENs, and multiple users with time-dependent requests is considered. The problem is formulated as a Stackelberg game, in which the SP (modeled as the leader) decides the prices for the storage and backhaul resources of the ENs (modeled as the followers). Since the entire game is a non-convex mixed-integer problem, which is even hard to solve in a centralized way, the problem is decomposed into two types of sub-games, a storage allocation game (SAG) and a number of user allocation games (UAGs). All of the sub-games are convex. We apply a Stackelberg game-based alternating direction method of multipliers (ADMM) to solve both the SAG and the UAGs in a distributed manner [21]–[23]. The main contributions of this work are listed as follows.

1) We prove that the convergence speed of the Stackelberg game-based ADMM is linearly related to the network size, measured by the number of ENs, the number of users, and the time frame within which users change their file demands.
2) Through analytical and simulation results, we show that the Stackelberg game-based ADMM can approximate the Stackelberg equilibrium, which requires fewer backhaul resources than centralized popularity-based caching and random caching approaches.
3) Based on both analytical and simulation results, we show that the total number of scalars exchanged between the SP and the ENs increases linearly relative to the network size. Therefore, the framework can potentially address large-scale caching problems.

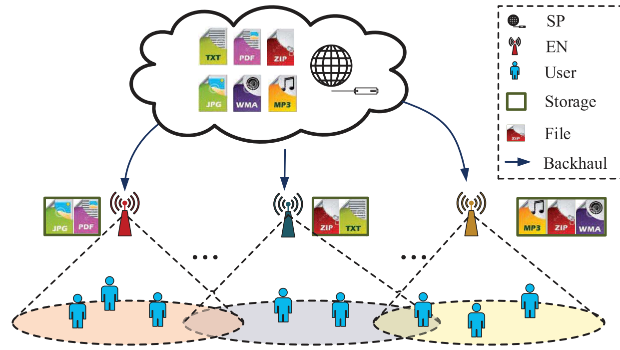


Fig. 1. System model.

The rest of this paper is organized as follows. In Section II, we introduce the system model and formulate the problem. The Stackelberg game-based ADMM is presented in Section III. In Section IV, the properties of the Stackelberg game-based ADMM are analyzed. Simulation results are provided in Section V. In Section VI, we draw our main conclusions.

II. SYSTEM MODEL

Consider an edge network as shown in Fig. 1, with one SP, K ENs with storage capability, and N users. The numbers of ENs and users can be very large in the future wireless networks. The SP generates a library of multimedia files, denoted by $\mathcal{F} = \{1, 2, \ldots, f, \ldots, F\}$. Each user n requires a set of files in the library from the SP. In traditional wireless networks without caching, when user n requires a file, f, from the SP, the SP first sends file f from the core network via a backhaul link to EN k that servers user n [3]. Then, EN k transmits file f to user n.

When the SP can anticipate the file demands of the users using, say, a learning mechanism [4], the SP proactively caches some popular files in the ENs so that these files can be directly transmitted to the users without going through a backhaul link when required. The above process is called proactive caching and can provide faster multimedia services and relieve the transmission burden on the backhaul links.

In the rest of this section, we present models of the distribution and coverage of the ENs in detail, the file demands from various users, the traffic model, and the time span. Then, we formulate and simplify the optimization problem of interest.

A. Distribution and Coverage of ENs

As in Fig. 2, we assume that the ENs are deployed in a two-dimensional $L \times W$ rectangular area following a homogeneous PPP [24]. That is to say, the location of each EN is drawn from an independent and uniform distribution in this

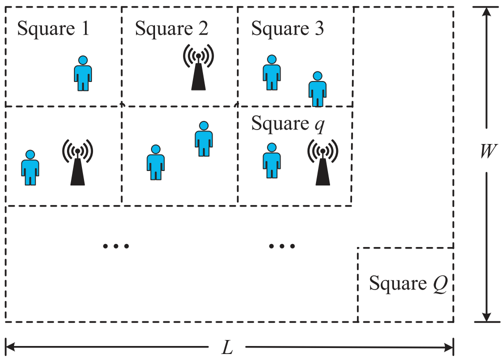


Fig. 2. Squares generating file demands.

rectangular area. Similarly, the locations of the users satisfy another independent PPP, that is, each user is independently and uniformly located in this area. Each EN k can transmit files to the users inside a circle area with a radius of d_0, which is also called the coverage of EN k. We set $I_{k,n} = 1$ if user n is inside EN k's coverage, otherwise $I_{k,n} = 0$. Define the converge matrix as $\boldsymbol{\xi} \in \mathbb{R}^{K\times N}$, where $\xi_{k,n} = 1$ if user n is served by EN k, otherwise $\xi_{k,n} = 0$. In this paper, we assume that each user can connect to exactly one EN. Hence, for each user n, the following constraint is naturally satisfied:

$$\sum_{k=1}^{K} I_{k,n}\xi_{k,n} = 1. \tag{1}$$

B. File Demand and Caching Capacity

As in Fig. 2, the large-scale network is divided into Q squares. In practice, the whole area may be divided into irregular sub-areas rather than the simple squares. In this paper, we use Q squares to simply simulate the change of demand probabilities, depending on real historical data.

Inside each square q, the demand probability for file f by user n, denoted by $p_{f,n}$, follows a Zipf (power-law) distribution [4],

$$p_{f,n} = \min\left\{\rho \cdot \frac{\Lambda_{q,f}^{-\kappa}}{\sum_{i=1}^{F}(i)^{-\kappa}}, 1\right\}, \tag{2}$$

where $\kappa > 0$ is the skewness parameter and can be anticipated by the SP, ρ describes the users' demand degree, where a larger ρ indicates that each user wants more files,[1] and $\boldsymbol{\Lambda}_q$ is a permutation of the vector $\{1, 2, \ldots, F\}$ for area q with $\Lambda_{q,f}$ being the f^{th} element of $\boldsymbol{\Lambda}_q$. $\boldsymbol{\Lambda}_q$ indicates that in different areas, the demand probabilities for different files can be different.

Denote by s_f the size of file f, which follows a uniform distribution between s_L and s_H. Denote by S_k^C the storage capacity of EN k. We use the file matrix $\boldsymbol{\eta} \in \mathbb{R}^{K\times F}$ to represent the set of files stored by the ENs, where $\eta_{k,f} = 1$ if file f is cached in EN k, otherwise $\eta_{k,f} = 0$. Therefore, the overall file size cached in EN k will be $S_k = \sum_{f=1}^{F}\eta_{k,f}\cdot s_f$ and must satisfy a capacity constraint,

$$S_k = \sum_{f=1}^{F}\eta_{k,f}\cdot s_f \le S_k^C. \tag{3}$$

C. Traffic Model

If file f requested by user n is cached in EN k, it can be transmitted from EN k directly. If it is not cached, EN k needs to set up a backhaul connection to the core network, i.e., the SP, which will allocate bandwidth $b_{k,n,f}$ to user n at EN k to transmit file f. It is natural to assume that each user n has a minimum bandwidth b_0 to satisfy a minimum backhaul download speed, that is,

$$b_{k,n,f} \ge b_0. \tag{4}$$

In this paper, we consider only the backhaul resources and not the spectrum and power resources in the traffic model. The scheduling of other resources is considered in [4], [7], and [8].

D. Time Span

Consider an one-day time span for the caching scenario. At off-peak times, e.g., late at night, when only a few users have file demands, the SP optimizes the storage allocation, that is, decides which files are cached at each EN. During peak times, i.e., in the daytime and evening, the SP optimizes which users access which ENs, i.e., user allocation, to consume the minimum total backhaul resources. The operation of caching from the SP to the ENs is operated at one-day intervals. In contrast, the user allocation is processed over a shorter time interval. Correspondingly, we assume that the locations and the file demands of users between two nearest user allocation processes are not changed. Thus, the SP's objective is to minimize the total backhaul resources for all users' demands throughout one day, denoted by U, through optimizing storage allocation and user allocation, given as

$$\begin{aligned}\min_{\boldsymbol{\eta},\boldsymbol{\xi}} \; U &= \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{f=1}^{F} B_k^{(t)} \\ &= \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{f=1}^{F} b_0(1-\eta_{k,f})\left(\sum_{n=1}^{N}\xi_{k,n}^{(t)}C_{n,f}^{(t)}\right),\end{aligned} \tag{5}$$

where $B_k^{(t)}$ is the total backhaul bandwidth for EN k, $C_{n,f}^{(t)}$ is an indicator for user n to request file f at time t, with $C_{n,f}^{(t)} = 1$ if requested and $C_{n,f}^{(t)} = 0$ otherwise, and T denotes the total number of times to perform the user allocation during one day. $T = 24$ indicates that the users change their file requests and locations hourly. $\mathbf{A}^{(t)}$ denotes the value of variable $\mathbf{A}$ at time t. Changes in users' locations and requests can be modeled as below.

At a time t, the system will generate independent matrices $\{I_{k,f}^{(t)}\}$ in (1), which indicates that the users can

[1] The range of the parameter ρ can be selected between 0 and $\frac{\sum_{i=1}^{F}(i)^{-\kappa}}{F^{-\kappa}}$. When $\rho = 0$, the demand probability of user n for file f, $p_{f,n} = 0$. This indicates no file is required from any user. When $\rho = \frac{\sum_{i=1}^{F}(i)^{-\kappa}}{F^{-\kappa}}$, for any user n and file f, the demand probability equals 1, which indicates that each user requires all files. When $\rho > 0$ and $\rho < \frac{\sum_{i=1}^{F}(i)^{-\kappa}}{F^{-\kappa}}$, for each user, the demand probability for different files can be different and vary between 0 and 1.

change their locations within their corresponding ENs' coverage randomly and independently. Then, the demand probability, $p_{f,n}^{(t)}$, in (2) will change with time, independently from square to square. Finally, based on the demand probability, $p_{f,n}^{(t)}$, the specific demand of each user, $C_{n,f}^{(t)}$, will be generated at time t. The time-relevant movement model or the time-relevant file demand model is not considered in this paper.

E. Problem Formulation

The authority to adjust the storage size, S_k, and the amount of the backhaul resources, B_k, are controlled by each EN k rather than the SP [25]. If the SP and ENs belong to different operators, the ENs are reluctant to help the SP without any incentives. Thus, the SP needs to incentivize each EN to allocate its storage and backhaul resources through payments. Then, each EN k's objective is to maximize its utility, denoted by V_k, calculated as follows:

$$\max_{\{\eta_{k,f}\}^{1\times F},\{\xi_{k,n}\}^{1\times N}} V_k = (\theta_k S_k - X_k(S_k)) + \sum_{t=1}^{T}\left(\phi_k^{(t)} B_k^{(t)} - Y_k(B_k^{(t)})\right), \quad (6)$$

where θ_k is the price to EN k for providing storage S_k, $X_k(S_k)$ represents the storage cost for EN k, $\phi_k^{(t)}$ is the price to EN k for providing $B_k^{(t)}$ backhaul bandwidth, and $Y_k(B_k^{(t)})$ represents the backhaul cost.

According to the law of diminishing marginal utility [26], both the cost functions $X_k(S_k)$ and $Y_k(B_k^{(t)})$ should be monotone increasing concave functions. For simplicity, we use quadratic functions, given by

$$X_k(S_k) = \alpha_k \cdot (S_k)^2, \quad (7)$$
$$Y_k(B_k^{(t)}) = \beta_k \cdot (B_k^{(t)})^2, \quad (8)$$

where $\alpha_k > 0$ and $\beta_k > 0$ are coefficients of proportionality. We briefly explain $X_k(s_k)$ and $Y_k(B_k^{(t)})$ in Remark 1.

Remark 1: The monotone increasing concavity is motivated mainly by the economics rather than the physical-layer properties. In practice, when there are other SPs in the network, such as in [16]–[18], EN k will call back storage resources from other SPs to provide services for the SP considered here. A rational EN k will take back the storage that caches files with lower popularity at first and then those with higher popularity. Thus, the number of served users will decrease faster when EN k calls back more storage resources. That is why the cost function, $X_k(S_k)$, is usually a monotone increasing function. In practice, $X_k(S_k)$ might be more complicated, such as in [16]–[18], but this feature is beyond the focus of this paper. Thus, we use the quadratic function here to reflect the monotone increasing concavity of the cost function. Similar analysis can be done for $Y_k(B_k^{(t)})$.

Based on the above discussion, the problem can be formulated as a Stackelberg game [27] with private information from the ENs, where the SP is the leader and the ENs are the K followers. In a Stackelberg game, the leader selects its strategy to optimize its utility first and then the followers move to optimize their utilities based on the leader's strategy. In particular, here the SP adjusts the prices, $\{\theta_k\}$ and $\{\phi_k^{(t)}\}$, as a strategy, to incentivize the ENs to cache the proper files, $\{\eta_{k,f}\}$, and serve the users, $\{\xi_{k,n}^{(t)}\}$, which are the followers' strategies. Mathematically, the Stackelberg game can be expressed as,[2]

$$\begin{aligned}
&\text{LG:}\quad \{\theta_k^\star\}, \left\{\left(\phi_k^{(t)}\right)^\star\right\} = \underset{\{\theta_{k,f}\},\left\{\phi_{k,n}^{(t)}\right\}}{\operatorname{argmin}} \; U,\\
&\text{FG:}\quad \{\eta_{k,f}^\star\}, \left\{\left(\xi_{k,n}^{(t)}\right)^\star\right\} = \underset{\{\eta_{k,f}\},\left\{\xi_{k,n}^{(t)}\right\}}{\operatorname{argmax}} \; V_k,\\
&\text{Constraints:}\quad (1)\ and\ (3),
\end{aligned} \quad (9)$$

where LG and FG denote the leader's game and the followers' game, respectively, and $\mathbf{A}^\star$ represents the optimal value of $\mathbf{A}$ when LG and FG reach a Stackelberg equilibrium.

F. Problem Decomposition

It is hard to reach a strict optimum in the game in (9) for both the SP and the ENs due the following reasons.

1) The storage allocation variables, $\{\eta_{k,f}\}$, and user allocation variables, $\left\{\xi_{k,n}^{(t)}\right\}$, are coupled in both the SP's utility in (5) and each EN's utility in (6). In this situation, the SP's utility in (5) is in general non-convex. Thus, it is hard to directly find the optimal values of $\{\eta_{k,f}\}$ and $\left\{\xi_{k,n}^{(t)}\right\}$ simultaneously.
2) Furthermore, $\{\eta_{k,f}\}$ and $\left\{\xi_{k,n}^{(t)}\right\}$ are 0-1 integer variables whereas the prices, $\{\theta_k\}$ and $\left\{\phi_k^{(t)}\right\}$, are continuous variables, and thus the game in (9) is a non-convex mixed integer problem, for which there is no general method to find an optimum [28].

To facilitate its solution and maintain low complexity, we decompose the game into a number of subgames as follows.

We decompose the game into $T+1$ subgames: a storage-allocation game and T user-allocation games. The SP and the ENs solve the SAG once and solve the UAG at each time t. Through the decomposition, all utility functions and the constraints in both the SAG and the UAGs are convex.

1) In the SAG, the SP and ENs play a subgame on the storage allocation and the prices for the storage resources, i,e., they solve for the optimal values of $\{\eta_{k,f}\}$ and $\{\theta_k\}$. And thus, the SAG is given by

$$\begin{aligned}
\text{LG:}\quad \{\theta_k^\star\} &= \underset{\{\theta_k\}}{\operatorname{argmin}}\, U_{SAG}\\
&= \underset{\{\theta_k\}}{\operatorname{argmin}} \sum_{k=1}^{K}\sum_{f=1}^{F} b_0(1-\eta_{k,f})\left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right),\\
\text{FG:}\quad \{\eta_{k,f}^\star\} &= \underset{\{\eta_{k,f}\}}{\operatorname{argmax}}\, V_{SAG,k}\\
&= \underset{\{\eta_{k,f}\}}{\operatorname{argmax}}\, \theta_k S_k - X_k(S_k),\\
\text{Constraints:}\quad &(3),
\end{aligned} \quad (10)$$

[2]The SP is assumed to be cost-unaware, which indicates that the SP cares only about how to incentive the ENs to reach the minimum backhaul, but is not concerned about the payments. Mathematically, the prices are not considered in the SP's utility.

where $E_{k,f}^{(t)}$ is an estimate of the total number of users for file f at time t, U_{SAG} is the utility of the SP, and $V_{SAG,k}$ is the utility of EN k in the SAG. The estimation of $E_{k,f}^{(t)}$ is discussed in Appendix A. In the SAG, the SP calculates $E_{k,f}^{(t)}$ rather than the exact value of each $C_{f,n}^{(t)}$.

2) With the optimal storage allocation in the SAG, in the UAG at each time t, the SP and the ENs play a sub-game on user allocation and the prices for the backhaul resources, i.e., they solve for the optimal values of $\left\{\xi_{k,n}^{(t)}\right\}$ and $\left\{\phi_{k,n}^{(t)}\right\}$. And thus, the UAG at time t is given by

$$\begin{aligned}
\text{LG:} \quad \left\{\left(\phi_{k,n}^{(t)}\right)^{\star}\right\} &= \underset{\{\theta_k\}}{\operatorname{argmin}}\, U_{UAG} \\
&= \underset{\left\{\phi_{k,n}^{(t)}\right\}}{\operatorname{argmin}} \sum_{k=1}^{K}\sum_{f=1}^{F} b_0(1-\eta_{k,f}^{\star}) \\
&\quad \times \left(\sum_{n=1}^{N} \xi_{k,n}^{(t)} C_{n,f}^{(t)}\right), \\
\text{FG:} \quad \left\{\left(\xi_{k,n}^{(t)}\right)^{\star}\right\} &= \underset{\{\eta_{k,f}\}}{\operatorname{argmax}}\, V_{UAG,k} \\
&= \underset{\left\{\xi_{k,n}^{(t)}\right\}}{\operatorname{argmax}} \sum_{n=1}^{N} \phi_{k,n}^{(t)}\xi_{k,n}^{(t)} - Y_k(B_k^{(t)}), \\
\text{Constraints:} \quad &(1),
\end{aligned} \tag{11}$$

where U_{UAG} is the utility of the SP and $V_{UAG,k}$ is the utility of EN k in the UAG at time t. In the UAG at time t, the SP needs to know the exact value of each $C_{f,n}^{(t)}$. That is to say, for the user allocation at time t, each user must tell the SP which files it requires.

III. Stackelberg Game-Based ADMM

In this section, the Stackelberg game-based ADMM is adopted to solve both the SAG and UAGs in a distributed manner. In the Stackelberg game-based ADMM, the SP adjusts the prices iteratively. At each iteration, under the current prices from the SP, each EN feeds back the storage (or backhaul) resources to reach an optimal point of a well-designed *incentive function* through a convex optimization approach, such as ADMM. Then, the SP adjusts the prices in the next iteration and the ENs feed back the optimal resources with the new prices. With a few iterations, the Stackelberg game-based ADMM can converge to a Stackelberg equilibrium. The Stackelberg game-based ADMM is suitable to solve the game especially when the network includes large numbers of ENs and users [21]–[23]. In this section, we will discuss the Stackelberg game-based ADMM for the SAG and the UAGs, respectively.

A. Stackelberg Game-Based ADMM for the SAG

It can be directly observed that the SAG can be further decomposed into K independent sub-SAGs. Sub-SAG k for the SP and EN k, is given by,

$$\begin{aligned}
\text{LG:} \quad \theta_k^{\star} &= \underset{\theta_k}{\operatorname{argmin}} \sum_{f=1}^{F} b_0(1-\eta_{k,f})\left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right), \\
\text{FG:} \quad \{\eta_{k,f}^{\star}\} &= \underset{\{\eta_{k,f}\}}{\operatorname{argmax}}\, \theta_k S_k - X_k(S_k), \\
\text{Constraint:} \quad &\sum_{f=1}^{F} \eta_{k,f}\cdot s_f \le S_k^C.
\end{aligned} \tag{12}$$

The SP can play K sub-SAGs with K ENs independently in parallel.

In sub-SAG k, the operation of the Stackelberg game-based ADMM is an iterative process. Given the price from the SP to each EN k at each step p, $\theta_k^{(p)}$, EN k updates the storage allocation as $\{\eta_{k,f}\} = \{\eta_{k,f}^{(p)}\}$ to minimize its *incentive function*, $\Theta_k^{(p)}$, defined as follows:

$$\Theta_k^{(p)} = \sum_{f=1}^{F} b_0(1-\eta_{k,f})\left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right) + X_k(S_k) - \theta_k^{(p)} S_k. \tag{13}$$

The incentive function equals a part of the SP's utility function plus EN k's utility in (12), which can be physically interpreted as follows: EN k helps optimize the SP's utility with the price and simultaneously optimize EN k's own utility. That is to say, the SP must tell EN k an estimate of the requirements for the files, $\sum_{t=1}^{T} E_{k,f}^{(t)}$.

With the incentive function in (13) for each EN, the Stackelberg game-based ADMM for the sub-SAG is a two-tier iteration, where the process to achieve $\left\{\eta_{k,f}^{(p)}\right\}$ inside each step p is called the *inner loop*, and the one to update the price, θ_k, from step p to step $p+1$ is called the *outer loop*.

1) *Inner loop:* The inner loop is processed on EN k. Given the price from the SP, $\theta_k^{(p)}$, EN k finds the optimal storage allocation, $\left\{\eta_{k,f}^{(p)}\right\}$, to minimize the incentive function in (13) under the constraints in (12) with an ADMM iterative process [29], where m is the iteration step for the inner loop.

a) Sequential updates for $\{\eta_{k,f} | f = 1, 2, \ldots, F\}$ are given by

$$\begin{aligned}
\eta_{k,f}^{(p)}(m+1) = \underset{\eta_{k,f}^{(p)}}{\operatorname{argmin}}\, &\Theta_k^{(p)} - \lambda_k^{(p)}(m)\eta_{k,f}^{(p)} s_f \\
&+ \frac{\gamma}{2}\left|\sum_{g=1, g\neq f}^{F} \eta_{k,g}^{(p)}(\hat{m}) s_g + \eta_{k,f}^{(p)} s_f - S_0\right|^2,
\end{aligned} \tag{14}$$

for $f = 1, 2, \ldots, F$, $\gamma > 0$ is a damping factor, and $\hat{m} = m+1$ when $f < g$ and $\hat{m} = m$ when $f > g$, which is set to satisfy the constraints in (12). With this step, EN k will achieve a set of continuous storage allocation variables with price $\theta_k^{(p)}$. The update must be operated sequentially on $\left\{\eta_{k,f}^{(p)}\right\}$ rather than in parallel, which is the basic condition to guarantee the convergence of the inner loop [30].

b) The boundary for $\{\eta_{k,f}^{(p)}(m+1)|f = 1,2,\dots,F\}$ is given by

$$\eta_{k,f}^{(p)}(m+1)=\begin{cases}0, & \eta_{k,f}^{(p)}(m+1)\le 0,\\ \eta_{k,f}^{(p)}(m+1), & 0<\eta_{k,f}^{(p)}(m+1)\le 1,\\ 1, & \eta_{k,f}^{(p)}(m+1)>1,\end{cases} \tag{15}$$

which is the common process to restrict the variables, $\left\{\eta_{k,f}^{(p)}(m+1)\right\}$, between 0 and 1.

c) The dual update is given by

$$\lambda_k^{(p)}(m+1) = \lambda_k^{(p)}(m) - \gamma\left(\sum_{f=1}^{F}(\eta_{k,f}^{(p)}(m+1)s_f) - S_k^0\right), \tag{16}$$

for $k = 1,2,\dots,K$, which guarantees the convergence with ADMM.

d) Steps a), b), and c) are repeated until each $\eta_{k,f}^{(p)}$ and each $\lambda_k^{(p)}$ have no significant change. (The sequential update guarantees convergence.) The current total amount of storage resources for caching files by EN k, $S_k^{(p)}$, is calculated by

$$S_k^{(p)} = \sum_{i=1}^{F}\eta_{k,f}^{(p)}s_f. \tag{17}$$

2) *Outer loop:* After EN k achieves its current quota of storage resources to cache files, $S_k^{(p)}$, EN k feeds back its *marginal cost* to the SP for price update as

$$\theta_k^{(p+1)} = \frac{\mathrm{d}X_k(S_k^{(p)})}{\mathrm{d}S_k}. \tag{18}$$

The marginal cost can be interpreted as the current cost for EN k to provide an additional unit storage when it has already provided $S_k^{(p)}$ resources to the SP. It is rational to set the price equal to or larger than the marginal cost when the SP wants EN k to provide more storage resources. The outer loop stops when the EN cannot adjust its storage too much, which mathematically satisfies a primal stopping criterion,

$$\left|\Theta_k^{(p)} - \Theta_k^{(p-1)}\right| < \varepsilon_1. \tag{19}$$

When the outer loop stops at step p, the EN k has already determined its storage resources as $S_k^{(p)}$ and the SP has determined price $\theta_k^{(p)}$. However, some variables in $\left\{\eta_{k,f}^{(p)}\right\}$ may not equal 0 or 1, which is not reasonable since EN k cannot cache only a part of a file. In this case, EN k must transform continuous variables in $\left\{\eta_{k,f}^{(p)}\right\}$ to discrete ones through sequencing: EN k caches F_k files with the largest $\eta_{k,f}^{(p)}$, where the constraints in (12) are simultaneously satisfied.

With K sub-SAGs, the SP and ENs determine which files are cached on which ENs, $\{\eta_{k,f}^{\star}\}$, and the optimal prices, $\{\theta_k^{\star}\}$.

B. Stackelberg Game-Based ADMM for the UAGs

The UAG is played at each time t. Different from that for the SAG, the SP can calculate the user allocation, $\{\xi_{k,n}^{(t)}\}$, in a centralized way. Then, the SP can ask each EN k to provide enough backhaul resources, B_k, to serve the allocated users with the Stackelberg game-based ADMM. The reasons are as follows.

1) As in the UAG in (11), each EN k cares only about its total backhaul resources, $B_k^{(t)}$, and the price, $\phi_k^{(t)}$, regardless of which users EN k serves. Thus, the SP can buy enough backhaul resources from EN k to serve the users.
2) The UAG in (11) does not have a constraint on private information of ENs, which is different from the SAG in (10) with a total storage constraint on EN k, S_k^C.
3) The centralized user allocation problem, as given below, has a closed-form solution for the SP, which does not need complicated calculations even when the network has a large number of users.

$$\min_{\left\{\xi_{k,n}^{(t)}\right\}} \sum_{k=1}^{K}\sum_{f=1}^{F} b_0(1-\eta_{k,f}^{\star})\left(\sum_{n=1}^{N}\xi_{k,n}^{(t)}C_{n,f}^{(t)}\right),$$
$$s.t. \sum_{k=1}^{K} I_{k,n}\xi_{k,n}^{(t)} = 1. \tag{20}$$

The closed-form solution for (20) is given by[3]

$$\xi_{k,n}^{(t)} = \begin{cases}1, & \chi_{k,n}^{(t)} = \min\{\chi_{l,n}^{(t)}|l=1,2,\dots,K\},\\ 0, & \chi_{k,n}^{(t)} > \min\{\chi_{l,n}^{(t)}|l=1,2,\dots,K\},\end{cases} \tag{21}$$

where $\chi_{k,n}^{(t)}$ represents the total backhaul resources needed by user n to obtain its desired files, calculated by,

$$\chi_{k,n}^{(t)} = \sum_{f=1}^{F} b_0(1-\eta_{k,f}^{\star})C_{n,f}^{(t)}. \tag{22}$$

For convenience, we use $\hat{\xi}_{k,n}^{(t)}$ to represent the user allocation calculated by the SP. From (20) and (21), the desired backhaul resources for each EN k, $\hat{B}_k^{(t)}$, can be calculated by

$$\hat{B}_k^{(t)} = \sum_{f=1}^{F} b_0(1-\eta_{k,f}^{\star})\left(\sum_{n=1}^{N}\hat{\xi}_{k,n}^{(t)}C_{n,f}^{(t)}\right). \tag{23}$$

Then, the UAG in (11) can be decomposed into K sub-UAGs at time t. Each sub-UAG for the SP and EN k is given by,

$$\text{LG:}\quad \left(\phi_k^{(t)}\right)^{\star} = \underset{\left\{\left(\phi_{k,n}^{(t)}\right)\right\}}{\operatorname{argmin}}\left(\frac{(B_k^{(t)})^2}{2\hat{B}_k^{(t)}} - B_k^{(t)}\right), \tag{24a}$$

$$\text{FG:}\quad B_{k^{(t)}} = \underset{B_k^{(t)}}{\operatorname{argmax}}\left(\phi_k^{(t)}\right)^{\star}B_k^{(t)} - Y_k\left(B_k^{(t)}\right), \tag{24b}$$

where $\left(\frac{(B_k^{(t)})^2}{2\hat{B}_k^{(t)}} - B_k^{(t)}\right)$ is designed by the SP to achieve $\hat{B}_k^{(t)}$ backhaul resources from EN k, since the optimal point of $\left(\frac{(B_k^{(t)})^2}{2\hat{B}_k^{(t)}} - B_k^{(t)}\right)$ is $B_k^{(t)} = \hat{B}_k^{(t)}$.

[3]When there are more than one $\chi_{k,n}^{(t)}$ that equal $\min\{\chi_{l,n}^{(t)}|l = 1,2,\dots,K\}$, e.g., $\chi_{k_1,n}^{(t)}$ and $\chi_{k_2,n}^{(t)}$, randomly select one index k, e.g., $k = k_1$ and let $\xi_{k_1,n}^{(t)} = 1$ and for other indices, e.g., k_2, let $\xi_{k_2,n}^{(t)} = 0$.

To incentive each EN k to provide $\hat{B}_k^{(t)}$ backhaul resources for the allocated users, the SP adopts the Stackelberg game-based ADMM to adjust the price, $\phi_k^{(t)}$. The operation of the Stackelberg game-based ADMM is an iterative process. Given the price from the SP to EN k at each step p, $\phi_k^{(t,p)}$, each EN k updates the backhual resources, $B_k^{(t)} = B_k^{(t,p)}$, to minimize its incentive function, $\Psi_k^{(t,p)}$, defined as,

$$\Psi_k^{(t,p)} = \left(\frac{(B_k^{(t)})^2}{2\hat{B}_k^{(t)}} - B_k^{(t)} \right) - \phi_k^{(t,p)} B_k^{(t)} + Y_k \left(B_k^{(t)} \right). \quad (25)$$

The first term is a part of the SP's utility in sub-UAG k in (24a), while the rest is EN k's utility in (24b). That is to say, the SP must tell EN k the desired backhaul resources, $\hat{B}_k^{(t)}$.

With the incentive function for each EN k, the Stackelberg game-based ADMM is an one-tier iteration described as follows.

1) With the price from the SP, $\phi_k^{(t,p)}$, EN k finds the optimal backhual bandwidth, $B_k^{(t,p)}$ to minimize the incentive function in (25). The closed-form solution for the incentive function minimization is given by

$$B_k^{(t,p)} = \frac{1 + \phi_k^{(t,p)}}{\frac{1}{\hat{B}_k^{(t)}} + 2\beta_k}. \quad (26)$$

Since the closed-form solution can be easily calculated, there is no need to use a numerical approach, such as ADMM.

2) After EN k achieves its current quota of backhaul resources to serve users, $B_k^{(t,p)}$, EN k feeds back its *marginal cost* to the SP, which is used to update the price as

$$\phi_k^{(t,p+1)} = \frac{\mathrm{d}Y_k(B_k^{(t,p)})}{\mathrm{d}B_k^{(t)}}. \quad (27)$$

The marginal cost can be interpreted as the current price for EN k to provide an additional unit of backhaul resources when EN k has already provided $B_k^{(p)}$ resources to the SP. It is reasonable to set the price equal to or larger than the marginal cost when the SP wants EN k to provide more backhaul resources.

3) The iteration stops at step p when each EN k cannot adjust its backhaul bandwidth too much any more, that is,

$$\left| \Psi_k^{(t,p)} - \Psi_k^{(t,p-1)} \right| < \varepsilon_2. \quad (28)$$

As a result, EN k provides $B_k^{(t,p)}$ backhaul resources, the set of users it serves is given in (21) and the price from the SP to EN k equals $\phi_k^{(t,p)}$.

The entire framework including the SAG and the UAGs is summarized in Algorithm 1.

Algorithm 1 Stackelberg game-based ADMM for SAG and UAG

SAG: The SP and each EN k play the sub-SAG in (12), $p = 0$
while $\Theta_k^{(p)} - \Theta_k^{(p-1)} \geq \varepsilon_1$ **do**
 (a) Optimize storage allocation with ADMM (inner loop) at EN k in (14)-(16);
 (b) EN k feeds back the marginal cost in (18) to the SP;
 (c) $p = p + 1$;
end
When the outer loop stops at step p, EN k caches F_k files with the largest $\eta_{k,f}^{(p)}$, where the constraint in (12) is simultaneously satisfied. The price from the SP equals $\theta_k^{(p)}$.
UAG: The SP and each EN k play the sub-UAG in (24) at each time t, $p = 0$
while $\Psi_k^{(p)} - \Psi_k^{(p-1)} \geq \varepsilon_2$ **do**
 (a) Optimize backhaul resources in closed-form at EN k in (26);
 (b) EN k feeds back the marginal cost in (27) to the SP;
 (c) $p = p + 1$;
end
EN k provides $B_k^{(t,p)}$ backhaul resources, and the set of users it serves is as in (21). The price from the SP to EN k equals $\phi_k^{(t,p)}$.

IV. Properties of Stackelberg Game-Based ADMM

In this section, we discuss the properties of the Stackelberg game-based ADMM for the SAG and UAGs. We first investigate the convergence and the optimum of the game. The total number of information exchanges is derived afterwards. Then, we discuss the scalability of our algorithm, and finally, the issue of outage is investigated.

A. Convergence

Based on [21, Proposition 1], the Stackelberg game-based ADMM for either the SAG or the UAGs converges at a linear rate. The rapid convergence of the SAG and the UAGs indicates that the SP and ENs do not spend too much time on the storage and backhaul resource allocation and price negotiation. Specifically, the average iteration time for the whole SAG in (10), denoted by $\bar{p}_{SAG} = \frac{\sum_{i=1}^{K} p_k}{K}$, is bounded by $O\left(\frac{NT}{\varepsilon_1}\right)$.[4] ε_1 describes the distance between the current value of the leader's utility function and the optimum, which is the primal stopping criterion given in (19). The reciprocal of ε_1 has a linear relation to the iteration, which corresponds to linear convergence (also known as first order convergence) [28]. The iteration time for the SAG also has a linear relation to the number of users, N, and to the time for backhual resource allocation, T. Similarly, the average iteration time for the UAGs at time t in (11), denoted by $\bar{p}_{UAG} = \frac{\sum_{i=1}^{K} p_k}{K}$, is bounded by $O\left(\frac{N}{\varepsilon_2}\right)$.

[4] $O(\cdot)$ represents the complexity. A function $p(N) = O(N)$ indicates that at least one $0 < p_0 < +\infty$ exists to guarantee $p(N) \leq p_0 \cdot N$. $O(N)$ is usually defined as linear complexity, $O(N^{\alpha})$ defines sublinear complexity when $0 < \alpha < 1$. $O(N^{\alpha})$ defines superlinear complexity when $\alpha > 1$ [31].

B. Optimum

In this part, we discuss the optima of the SAG and UAGs, respectively.

1) Approximate Optimum of the SAG: The SAG will converge into an approximate Stackelberg equilibrium, in which the utilities of the SP and each EN approximate the optimal utility values in the SAG with storage resource allocation and price negotiation. The gaps to the optimal utilities for the SP and each EN are listed in Theorems 1 and 2, proved in Appendices B and C, respectively.

Theorem 1: Denote the optimal utility value of the SP in the SAG in (10) as $U^{\star}_{SAG}$. When the Stackelberg game-based ADMM converges, the utility of the SP is denoted by $U^{(p)}_{SAG}$. Then, the maximum gap between $U^{(p)}_{SAG}$ and $U^{\star}_{SAG}$ is given by

$$\begin{aligned} 0 \leq U^{(p)}_{SAG} - U^{\star}_{SAG} &\leq b_0 \cdot \sum_{k=1}^{K} \left(\max_{f} \sum_{t=1}^{T} E^{(t)}_{k,f} \right) \\ &\leq K b_0 \max_{k,f} \sum_{t=1}^{T} E^{(t)}_{k,f} = O(KNT). \end{aligned} \tag{29}$$

Theorem 1 indicates that the Stackelberg game-based ADMM has at worst a K-file gap to the minimum utility of the SP. More specifically, the Stackelberg game-based ADMM fails to cache at most one file in each EN.

Theorem 2: Denote the optimal utility value of each EN k in the SAG in (10) as $V^{\star}_{SAG,k}$. When the Stackelberg game-based ADMM converges, the utility of EN k is denoted by $V^{(p)}_{SAG,k}$. Then, the maximum gap between $V^{(p)}_{SAG,k}$ and $V^{\star}_{SAG,k}$ is given by,

$$0 \leq V^{\star}_{SAG,k} - V^{(p)}_{SAG,k} \leq \alpha_k (s_H)^2. \tag{30}$$

Theorem 2 indicates that the Stackelberg game-based ADMM reaches at worst a one-file lower utility of EN k, $\alpha_k(s_H)^2$.

2) Strict Optimum of the UAG at Time t: The UAG at time t will converge into a strict Stackelberg optimum, in which the utilities of the SP and each EN reach the strict optimal utility values in the UAG at time t. Specifically, the optimal utilities for the SP and each EN are listed in Corollary 1 and Corollary 2, which can be directly derived based on [21, Proposition 1], [22, Proposition 1], and [23, Remark 6].

Corollary 1: When the Stackelberg game-based ADMM converges for all sub-UAGs in (24) at the maximum step p, the utility of the SP, denoted by $U^{(p)}_{UAG}$, reaches the strictly optimal point, in which $U^{(p)}_{UAG}$ is minimized.

Corollary 2: When the Stackelberg game-based ADMM converges for each sub-UAG k in (24) at step p, the utility of EN K, denoted by $V^{(p)}_{UAG,k}$, reaches the strictly optimal point, at which $V_{UAG,k}$ is maximized.

3) Suboptimum for the Entire Game: The entire game in (9) can reach a suboptimal point through implementing the Stackelberg game-based ADMM in the SAG and the T UAGs. However, the gap between the optimal utility values and our algorithm is hard to find even for a strictly optimal solution found in a centralized manner.

C. Information Exchange

The total amount of information exchange can be described by the number of scalars exchanged between the SP and the ENs. In the SAG, each EN k needs to send the marginal cost, $\frac{\mathrm{d}X_k(S_k^{(p)})}{\mathrm{d}S_k}$, and the storage, $S_k^{(p)}$, to the SP at each step in the outer loop. The total number of scalars from K ENs sent to the SP equals $2K$. Then, the SP will transmit the new prices, $\{\theta_k^{(p+1)}\}$, to K ENs. Finally, when the Stackelberg game-based ADMM for each sub-SAG converges at step $O(\frac{NT}{\varepsilon_1})$, the SP needs to send the final storage allocation results, $\{\eta^{\star}_{k,f}\}$, KF scalars, to the ENs. Therefore, the total number of scalars transmitted between the SP and the ENs in the SAG is given by

$$\Gamma_S = 3K \cdot O\left(\frac{NT}{\varepsilon_1}\right) + KF. \tag{31}$$

The total amount of information exchange for the UAG at each time t can be calculated similarly. For the UAG throughout one day, $t = \{1, 2, \ldots, T\}$, the total number of scalars transmitted between the SP and the ENs is given as follows:

$$\Gamma_U = T\left(3K \cdot O\left(\frac{N}{\varepsilon_2}\right) + KN\right). \tag{32}$$

Thus, we have the corollary on the total information exchange as below.

Corollary 3: The total number of transmitted scalars in the entire game, Γ, is given by

$$\begin{aligned} \Gamma &= \Gamma_S + \Gamma_U \\ &= 3K \cdot O\left(\frac{NT}{\varepsilon_1}\right) + KF + T\left(3K \cdot O\left(\frac{N}{\varepsilon_2}\right) + KN\right) \\ &= O(KNT). \end{aligned} \tag{33}$$

D. Scalability

Scalability indicates how the algorithm performs when the network size increases. It is very important to extend an algorithm to large-scale networks such that the performance metrics for the algorithm have a linear relation to, sub-linear relation to, or are independent of the network size [32]. In this paper, the metrics that describe our algorithm performance are the convergence speed in Section IV-A, the optimum gap in Section IV-B, and the information exchange in Section IV-C. The variables that describe the network size are the number of ENs K, the number of users N, and the time for use allocation, T. The metrics that can be analytically derived have been given in the above sections. The others will be examined through simulation results. In a word, the optimum can be guaranteed and the convergence speed is linear with or independent of the network size. This indicates that our algorithm is scalable for future large scale networks.

E. Outage

Besides the total backhual resources, the outage is another important performance metric in caching problems [13]. Outage is defined as the proportion of file requests that cannot

TABLE I
SIMULATION PARAMETERS

$L \times W$	Constant 1000m×1000m
K	Variable between 10 and 100
N	Variable between 100 and 1000
F	Constant 20
ρ	Constant 1
Q	Constant 100
T	Variable between 1 and 24
b_0	Constant 5 Units
S_k^C	Constant 200 Units
s_f	Uniformly distributed between 10 and 30 Units
α_k	Uniformly distributed between 0 and 10^{-4}
β_k	Uniformly distributed between 0 and 10^{-4}
ε_1	Constant 0.1
ε_2	Constant 0.1

be fulfilled by the ENs directly throughout the entire network. Mathematically, the outage can be calculated as follows:

$$\Omega = \frac{\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{f=1}^{F}(1-\eta_{k,f})\left(\sum_{n=1}^{N}\xi_{k,n}^{(t)}C_{n,f}^{(t)}\right)}{\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{f=1}^{F}\left(\sum_{n=1}^{N}\xi_{k,n}^{(t)}C_{n,f}^{(t)}\right)}. \quad (34)$$

From (34), the outage is obviously linearly dependent on the total amount of the backhaul resources, i.e., the SP's utility in (5). Thus, we omit any further discussion of this metric.

V. SIMULATION RESULTS

Several representative examples are shown through simulation results, whose parameters are listed in TABLE I. The simulation results include three aspects: the convergence, the optimum, and the total amount of information exchange. The independence or the approximate linear relation between the convergence speed and the network size indicates that the Stackelberg game-based ADMM can be applied in large scale networks. The optimum indicates that the Stackelberg game-based ADMM can approximately achieve the minimum of the total backhaul resources no matter how the network size changes. Finally, we show that the total number of scalars that the SP and the ENs exchange in the Stackelberg game-based ADMM has a linear correlation with the network size.

A. Convergence

In this part, we illustrate the convergence of the SAG and the UAGs in Fig. 3 and Fig. 4, respectively. In Fig. 3, the relations between the average iteration time, $\bar{p}_{SAG}$, in the SAG and the number of ENs, K, the number of users, N, and the time for user allocation, T, are provided. From the figure, $\bar{p}_{SAG}$ changes sublinearly with N and T. In addition, $\bar{p}_{SAG}$ is always below two, which indicates that the storage allocation and price negotiation between the SP and the K ENs can accomplished within two steps on average.

In Fig. 4, we illustrate the relations between the average iteration time, $\bar{p}_{UAG}$, in the UAG and the number of ENs, K, the number of users, N, and the time for user allocation, T. From Fig. 4, $\bar{p}_{UAG}$ is independent of K and T and increases with N at an approximately linear speed.

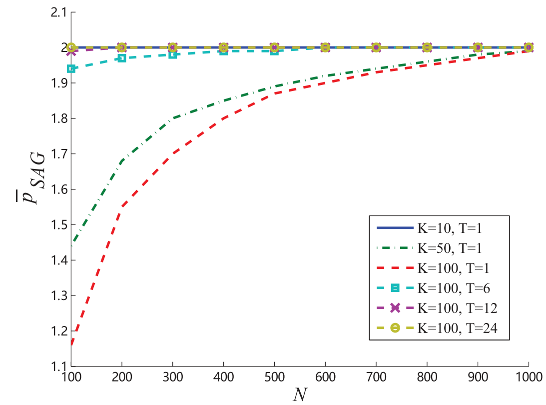


Fig. 3. Convergence of the SAG.

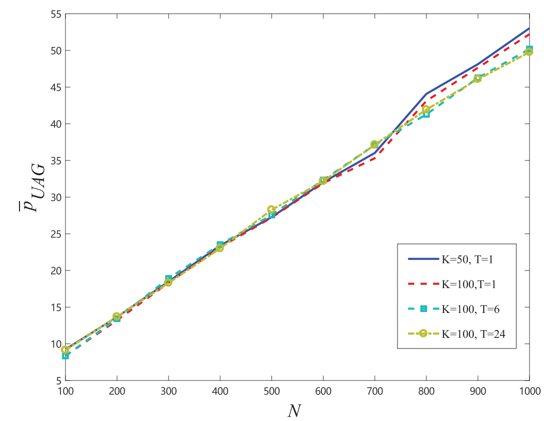


Fig. 4. Convergence of the UAGs.

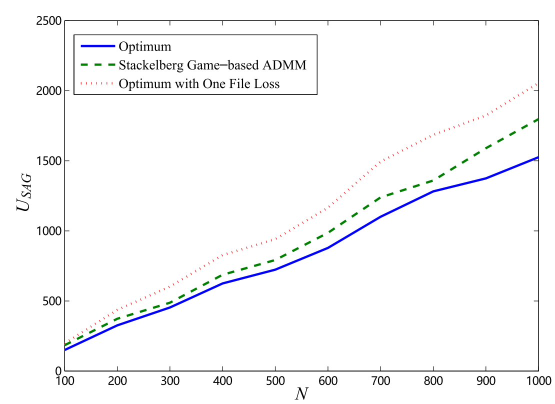


Fig. 5. The optimum of the SP's utility in the SAG, U_{SAG}, with $K = 100$.

B. Optimum

In Fig. 5 and Fig. 6, we show the approximate optimum of the SAG. In Fig. 5, the SP's utility in the SAG, U_{SAG}, can approximate the optimal one. The optimum is reached through relaxing the SAG to a continuous problem, and the one-file loss indicates that the optimal value plus a one-file backhual as in (29). The gap increases linearly with the number of users, N, as indicated by Theorem 1. In Fig 6, the average of the ENs' utilities in the SAG is also inside a one-file gap with the optimal value and is independent of the number of users as discussed by Theorem 2.

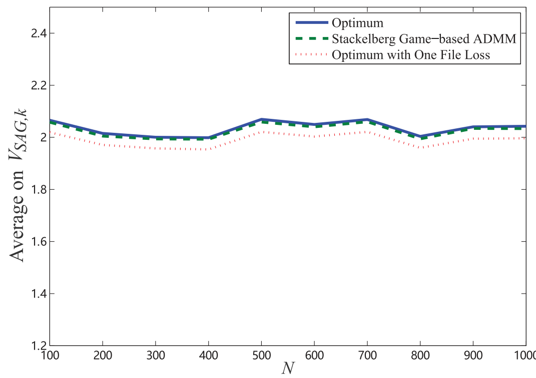


Fig. 6. The optimum of the average of the ENs' utilities in the SAG, $\bar{V}_{SAG,k}$, with $K = 100$.

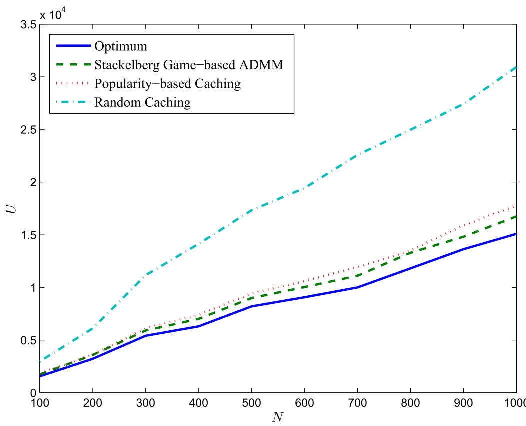


Fig. 7. Total backhaul resources, U, with the Stackelberg game-based ADMM, popularity-based caching, and random caching when $K = 100$, $T = 1$, and N varies.

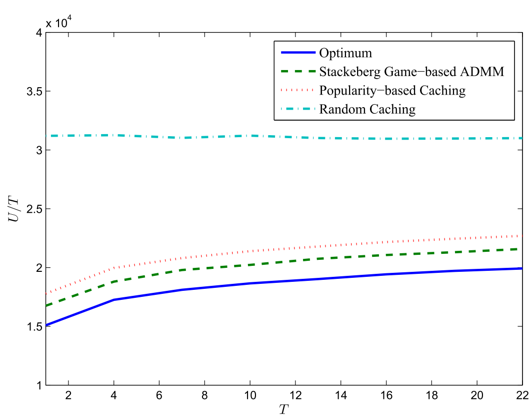


Fig. 8. Total backhaul resources, U, with the Stackelberg game-based ADMM, popularity-based caching, and random caching when $K = 100$, $N = 1000$, and T varies.

In Fig. 7 and Fig. 8, we compare the performance of the Stackelberg game-based ADMM with two other approaches. In popularity-based caching, each EN k caches the most popular files, i.e., the files with the largest $\sum_{t=1}^{T} E_{k,f}^{(t)}$, and the user accesses the EN to obtain as many files as possible at each time t. Similar approaches can be found in [4] assuming the users' demands follow Zipf's law. In random caching, each EN K caches files that can fill up the storage capacity.

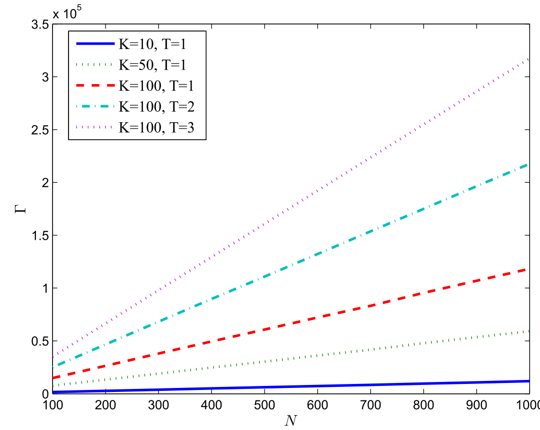


Fig. 9. The number of scalars transmitted in the entire game with the Stackelberg game-based ADMM when K, N, and T vary.

This usually happens when the users' file demands follow a uniform distribution. The optimum is reached through relaxing the SAG to a continuous problem, and then solving the SAG and UAGs in a centralized manner. We test the utility of the SP, U, for the entire game in (9), i.e., the total backhaul, and find that the our algorithm achieves a lower total backhaul than popularity-based caching and random caching. In Fig. 7, the total backhual increases approximately linearly with the number of users.

In Fig. 8, a larger time T means that the users' file demands change more frequently, in which case the caching process may miss more users' requests. Thus, the average of the total backhual at each time, U/T, increases with the time of user allocation.

C. Information Exchange

Fig. 9 shows the total number of scalars, Γ, that are transmitted in the entire game. It is can be observed that Γ has a linear correlation with the number of users, N. In addition, comparing different curves, Γ is approximately linearly correlated with the number of ENs, K, and the number for user allocations, T. This corroborates Corrollary 3.

VI. Conclusion

In this paper, we have proposed a convergent and scalable Stackelberg game for edge caching. Specifically, a network with one SP, multiple ENs, and multiple users with time-dependent requests has been considered. The problem has been formulated as a Stackelberg game, and decomposed into two types of sub-games, a SAG and a number of UAGs. The proposed Stackelberg game-based ADMM can solve both the SAG and UAGs in a distributed manner. In either the SAG or each UAG, the Stackelberg game-based ADMM converges linearly in the network size. In addition, it has been proved that the Stackelberg game-based ADMM approximates the optimum of the SAG to within a gap having a linear relation with the network size. The Stackelberg game-based ADMM has been proved to reach the strict optimum of each UAG. The entire game achieves lower total backhaul

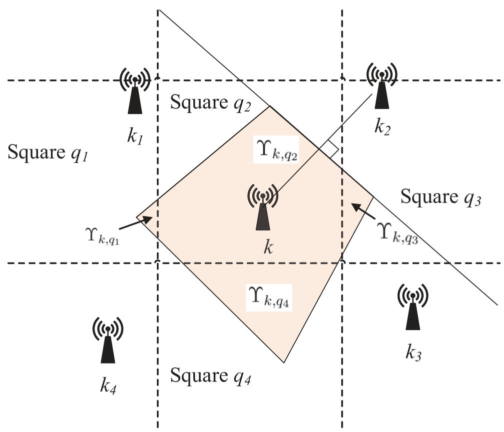


Fig. 10. Coverage estimation of the number of users by EN k.

than centralized popularity-based caching and random caching. Furthermore, based on our analytical and simulation results, the total number of scalars exchanged in the network has a linear correlation with the network size. The rapid convergence and scalability of our framework have shown its potential to address caching for networks with large numbers of ENs and users.

Appendix A
The Estimate $E_{k,f}^{(t)}$ in (10)

$E_{k,f}^{(t)}$ is the estimate of the total number of users that are allocated to EN k, for file f at time t. We calculate $E_{k,f}^{(t)}$ under the following assumption.

Assumption 1: Each user is willing to connect to the nearest EN.

Then, based on stochastic geometry, $E_{k,f}^{(t)}$ can be calculated as,

$$E_{k,f}^{(t)} = \mathbb{E}\left(\sum_{n=1}^{N} \xi_{k,n}^{(t)} C_{n,f}^{(t)}\right) = \frac{N}{L \cdot W} \cdot \sum_{q=1}^{Q}\left(\Upsilon_{k,q} \cdot \min\left\{\rho^{(t)} \cdot \frac{\left(\Lambda_{q,f}^{(t)}\right)^{-\kappa}}{\sum_{i=1}^{F}(i)^{-\kappa}}, 1\right\}\right), \tag{35}$$

where $\mathbb{E}$ denotes the expectation, $\Upsilon_{k,q}$ is the coverage of EN k in square q based on the rule that each user is likely to access the nearest EN. An example is shown in Fig. 10. The solid lines represent the coverage boundary for EN k. The dotted lines represent the boundaries of squares. Then, $E_{k,f}^{(t)}$ can be calculated from

$$E_{k,f}^{(t)} = \frac{N}{L \cdot W} \cdot \Upsilon_{k,q_1} \cdot \min\left\{\rho^{(t)} \cdot \frac{\left(\Lambda_{q_1,f}^{(t)}\right)^{-\kappa}}{\sum_{i=1}^{F}(i)^{-\kappa}}, 1\right\} + \frac{N}{L \cdot W} \cdot \Upsilon_{k,q_2} \cdot \min\left\{\rho^{(t)} \cdot \frac{\left(\Lambda_{q_2,f}^{(t)}\right)^{-\kappa}}{\sum_{i=1}^{F}(i)^{-\kappa}}, 1\right\}, + \frac{N}{L \cdot W} \cdot \Upsilon_{k,q_3} \cdot \min\left\{\rho^{(t)} \cdot \frac{\left(\Lambda_{q_3,f}^{(t)}\right)^{-\kappa}}{\sum_{i=1}^{F}(i)^{-\kappa}}, 1\right\} + \frac{N}{L \cdot W} \cdot \Upsilon_{k,q_4} \cdot \min\left\{\rho^{(t)} \cdot \frac{\left(\Lambda_{q_4,f}^{(t)}\right)^{-\kappa}}{\sum_{i=1}^{F}(i)^{-\kappa}}, 1\right\}. \tag{36}$$

Appendix B
Proof of Theorem 1

The minimum value of the SP's utility in the SAG in (10), $U_{SAG}^{\star}$, can be reached through solving K 0-1 integer programming problems as follows:

$$\min_{\{\eta_{k,f}\}} \sum_{f=1}^{F} b_0(1-\eta_{k,f})\left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right), \quad s.t. \sum_{f=1}^{F} \eta_{k,f} \cdot s_f \leq s_k^C, \tag{37}$$

where $k = 1, 2, \ldots, K$. Each problem in (37) a typical knapsack problem [32] when each $\eta_{k,f}$ is a 0-1 integer variable, which is NP-hard. We relax the problem in (37) to a continuous problem, denoted by Ξ_k, through continuation, in which each $\eta_{k,f}$ satisfies $0 \leq \eta_{k,f} \leq 1$. The optimal utility for the continuous problem, Ξ_k is denoted by $\hat{U}_{SAG,k}$. Then, it is intuitive that the following inequality is satisfied:

$$\sum_{k=1}^{K} \hat{U}_{SAG,k} \leq U_{SAG}^{\star}. \tag{38}$$

Then, we prove Theorem 1 through proving the following lemma.

Lemma 1: When the Stackelberg game-based ADMM for the sub-SAG k in (12) converges at step p, the following inequation is satisfied:

$$\sum_{f=1}^{F} b_0(1-\eta_{k,f}^{(p)})\left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right) - \hat{U}_{SAG,k} \leq \max_{f} b_0\left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right). \tag{39}$$

Proof: When the Stackelberg game-based ADMM for the sub-SAG k converges at step p, there exists at most one $\eta_{k,f}^{(p)}$ for index f such that $0 < \eta_{k,f}^{(p)} < 1$. For any other $\eta_{k,f}^{(p)}$ for index f, it equals either 0 or 1. The reasons for this are listed as follows.

1) First, based on a proof in our prior work in ([22], Proposition 1), the term $b_0(1-\eta_{k,f}^{(p)})\left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right)$ will reach a minimum point where each $\eta_{k,f}$ is a continuous variable.
2) Then, the step in (15) can restrict each $\eta_{k,f}^{(p)}$ to satisfy $0 \leq \eta_{k,f}^{(p)} \leq 1$.
3) Finally, when there are more than one $\eta_{k,f}^{(p)}$ for index f that do not equal 0 or 1, assume that $0 < \eta_{k,f_1}^{(p)} < 1$ and $0 < \eta_{k,f_2}^{(p)} < 1$, and $\frac{\sum_{t=1}^{T} E_{k,f_1}^{(t)}}{s_1} \leq \frac{\sum_{t=1}^{T} E_{k,f_2}^{(t)}}{s_2}$. Then we can reach a smaller value of $b_0(1-\eta_{k,f}^{(p)})\left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right)$

when η_{k,f_1} subtracts δ and η_{k,f_2} adds $\frac{\delta s_1}{s_2}$, where $\delta > 0$ is a small scalar. This deviates from the property in 1). Hence, we have at most one $\eta_{k,f}^{(p)}$ for index f such that $0 < \eta_{k,f}^{(p)} < 1$.

When $\eta_{k,f}^{(p)} = 1$ for index f, file f is cached by EN k. When $\eta_{k,f}^{(p)} = 0$, file f is not cached. When $0 < \eta_{k,f}^{(p)} < 1$, whether it is cached depends on whether EN k has enough storage resources. The worst case is that file f is always not cached when $0 < \eta_{k,f}^{(p)} < 1$. Assuming that f_0 is the file with $0 < \eta_{k,f}^{(p)} < 1$. In this case, we have

$$\sum_{f=1}^{F} b_0(1-\eta_{k,f}^{(p)}) \left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right) - \hat{U}_{SAG,k} \leq b_0 \left(\sum_{t=1}^{T} E_{k,f_0}^{(t)}\right) \leq \max_f b_0 \left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right). \quad (40)$$

This completes the proof of Lemma 1.

Finally, we can derive that

$$\begin{aligned} U_{SAG}^{(p)} - U_{SAG}^{\star} &= \sum_{k=1}^{K}\sum_{f=1}^{F} b_0(1-\eta_{k,f}^{(p)}) \left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right) - U_{SAG}^{\star} \\ &\leq \sum_{k=1}^{K} \left(\sum_{f=1}^{F} b_0(1-\eta_{k,f}^{(p)}) \left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right) - \hat{U}_{SAG,k}\right) \\ &\leq \sum_{k=1}^{K} \left(\max_f b_0 \left(\sum_{t=1}^{T} E_{k,f}^{(t)}\right)\right) \leq K b_0 \max_{k,f} \sum_{t=1}^{T} E_{k,f}^{(t)}. \end{aligned} \quad (41)$$

Since $E_{k,f}^{(t)}$ is intuitively proportional to the number of users, N, we have

$$K b_0 \max_{k,f} \sum_{t=1}^{T} E_{k,f}^{(t)} = O(KNT). \quad (42)$$

This completes the proof of Theorem 1.

APPENDIX C
PROOF OF THEOREM 2

Based on [22, Remark 6], the Stackelberg game-based ADMM can reach the strict optimal point for EN k in sub-SAG k in (12) when each $\eta_{k,f}$ is a continuous variable. Correspondingly, the optimal value of the storage provided by EN k is denoted by $\hat{S}_{SAG,k}$, and the optimal value of EN k's utility is denoted by $\hat{V}_{SAG,k}$. When each $\eta_{k,f}$ is a 0-1 integer, the optimal value of the storage provided by EN k is denoted by $S_{SAG,k}^{\star}$. Intuitively, we have

$$\begin{aligned} V_{SAG,k}^{\star} &= \theta_k^{\star} S_{SAG,k}^{\star} - \alpha_k \left(S_{SAG,k}^{\star}\right)^2 \\ &\leq \hat{V}_{SAG,k} = \theta_k^{\star} \hat{S}_{SAG,k} - \alpha_k \left(\hat{S}_{SAG,k}\right)^2. \end{aligned} \quad (43)$$

Then, we prove Theorem 2 through proving the following lemma.

Lemma 2: When the Stackelberg game-based ADMM for sub-SAG k converges at step p, the following inequality is satisfied:

$$\hat{V}_{SAG,k} - V_{SAG,k}^{(p)} \leq \alpha_k (s_H)^2. \quad (44)$$

Proof: We have the following inequality:

$$\begin{aligned} &\hat{V}_{SAG,k} - V_{SAG,k}^{(p)} \\ &= \theta_k^{\star} \hat{S}_{SAG,k} - \alpha_k \left(\hat{S}_{SAG,k}\right)^2 - \left(\theta_k^{\star} S_{SAG,k}^{(p)} - \alpha_k \left(S_{SAG,k}^{(p)}\right)^2\right) \\ &\leq \theta_k^{\star} \hat{S}_{SAG,k} - \alpha_k \left(\hat{S}_{SAG,k}\right)^2 - \left(\theta_k^{\star}\left(\hat{S}_{SAG,k} - s_H\right) - \alpha_k \left(\hat{S}_{SAG,k} - s_H\right)^2\right) \\ &= s_H \left(\theta^{\star} - 2\alpha_k \hat{S}_{SAG,k} + \alpha_k s_H\right). \end{aligned} \quad (45)$$

According to [22, Remark 6], the strictly optimal point for EN k in sub-SAG k in (12) is reached when each $\eta_{k,f}$ is a continuous variable. Thus, we have

$$\frac{\partial \hat{V}_{SAG,k}}{\partial \hat{S}_{SAG,k}} = \theta^{\star} - 2\alpha_k \hat{S}_{SAG,k} = 0. \quad (46)$$

We combine (46) into (45) and finally achieve that

$$\hat{V}_{SAG,k} - V_{SAG,k}^{(p)} \leq \alpha_k (s_H)^2. \quad (47)$$

Based on Lemma 2, we can finally derive that

$$V_{SAG,k}^{\star} - V_{SAG,k}^{(p)} \leq \hat{V}_{SAG,k} - V_{SAG,k}^{(p)} \leq \alpha_k (s_H)^2. \quad (48)$$

This completes the proof of Theorem 2.

REFERENCES

[1] Cisco, “Visual networking index: Global mobile data traffic forecast update, 2016–2021,” Cisco, San Jose, CA, USA, White Paper, Mar. 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[2] Z. Han, M. Hong, and D. Wang, *Signal Processing and Networking for Big Data Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

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

[4] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: The role of proactive caching in 5G wireless networks,” *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[5] 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.

[6] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, “FemtoCaching: Wireless video content delivery through distributed caching helpers,” in *Proc. INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 1107–1115.

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

[8] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, “Multicast-aware caching for small cell networks,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Istanbul, Turkey, Apr. 2014, pp. 2300–2305.

[9] 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.

[10] M. Ji, G. Caire, and A. F. Molisch, “Wireless device-to-device caching networks: Basic principles and system performance,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.

[11] K. Poularakis and L. Tassiulas, “Exploiting user mobility for wireless content delivery,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013, pp. 1017–1021.

[12] C. Yang, Y. Yao, Z. Chen, and B. Xia, “Analysis on cache-enabled wireless heterogeneous networks,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 131–145, Jan. 2016.

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

[14] Y. Chen, M. Ding, J. Li, Z. Lin, G. Mao, and L. Hanzo, "Probabilistic small-cell caching: Performance analysis and optimization," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 4341–4354, May 2017.

[15] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1024–1036, Feb. 2017.

[16] Z. Hu, Z. Zheng, T. Wang, and L. Song, "Caching as a service: Small-cell caching mechanism design for service providers," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6992–7004, Oct. 2016.

[17] J. Li, H. Chen, Y. Chen, Z. Lin, B. Vucetic, and L. Hanzo, "Pricing and resource allocation via game theory for a small-cell video caching system," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2115–2129, Aug. 2016.

[18] F. Shen, K. Hamidouche, E. Bastug, and M. Debbah, "A Stackelberg game for incentive proactive caching mechanisms in wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[19] T. Liu, J. Li, F. Shu, and Z. Han, "Resource trading for a small-cell caching system: A contract-theory based approach," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.

[20] Y. Tan, Y. Yuan, T. Yang, Y. Xu, and B. Hu, "Femtocaching in wireless video networks: Distributed framework based on exact potential game," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Chengdu, China, Jul. 2016, pp. 1–6.

[21] Z. Zheng, L. Song, and Z. Han, "Bridge the gap between ADMM and Stackelberg game: Incentive mechanism design for big data networks," *IEEE Signal Process. Lett.*, vol. 24, no. 2, pp. 191–195, Feb. 2017.

[22] Z. Zheng, L. Song, and Z. Han, "Bridging the gap between big data and game theory: A general hierarchical pricing framework," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[23] Z. Zheng, L. Song, Z. Han, G. Y. Li, and H. V. Poor, "Multi-leader multi-follower game-based ADMM for big data processing," in *Proc. 18th IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sapporo, Japan, Jul. 2017, pp. 1–5.

[24] J. G. Andrews, F. Baccelli, and R. K. Ganti, "A tractable approach to coverage and rate in cellular networks," *IEEE Trans. Commun.*, vol. 59, no. 11, pp. 3122–3134, Nov. 2011.

[25] Z. Hu, Z. Zheng, T. Wang, L. Song, and X. Li, "Game theoretic approaches for wireless proactive caching," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 37–43, Aug. 2016.

[26] N. G. Mankiw, *Principles of Microeconomics*, 8th ed. New York, NY, USA: Worth Publishers, 2012.

[27] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Nov. 2010.

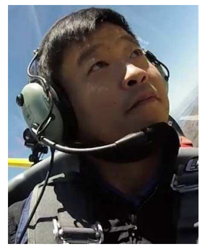
[30] T. Lin, S. Ma, and S. Zhang, "On the sublinear convergence rate of multi-block ADMM," 2015. [Online]. Available: https://arxiv.org/abs/1408.4265

[31] S. Yan, *Number Theory for Computing*. Berlin, Germany: Springer, 2002.

[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.

Zijie Zheng (S'14) received the B.S. degree in electronic engineering from Peking University, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Electronics Engineering and Computer Science. His current research interests include game theory and optimization in 5G networks, wireless powered networks, mobile social networks, and wireless big data.

Lingyang Song (S'03–M'06–SM'12) received the Ph.D. degree from the University of York, U.K., in 2007. He was a Research Fellow with the University of Oslo, Norway, until rejoining Philips Research, U.K., in 2008. In 2009, he joined the School of Electronics Engineering and Computer Science, Peking University, where he is currently a Boya Distinguished Professor. His main research interests include wireless communication and networks, signal processing, and machine learning. He was a recipient of the IEEE Leonard G. Abraham Prize in 2016 and the IEEE Asia Pacific Young Researcher Award in 2012. He received the K. M. Stott Prize for excellent research from the University of York. He has been an IEEE Distinguished Lecturer since 2015.



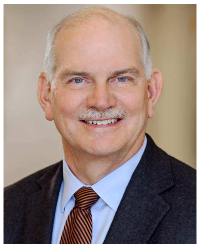
Zhu Han (S'01–M'04–SM'09–F'14) received the B.S. degree in electronic engineering from Tsinghua University in 1997 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively.

From 2000 to 2002, he was a Research and Development Engineer at JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an Assistant Professor at Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department and the Computer Science Department, University of Houston, Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He received an NSF CAREER Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in Communication Systems (Best Paper Award from the IEEE JSAC) in 2016, and several best paper awards at IEEE conferences. He is currently an IEEE Communications Society Distinguished Lecturer. He was a 1% Highly Cited Researcher in 2017 according to Web of Science.

Geoffrey Ye Li (S'93–M'95–SM'97–F'06) received the B.S.E. and M.S.E. degrees from the Department of Wireless Engineering, Nanjing Institute of Technology, Nanjing, China, in 1983 and 1986, respectively, and the Ph.D. degree from the Department of Electrical Engineering, Auburn University, Auburn, AL, USA, in 1994.

He was a Teaching Assistant and a Lecturer with Southeast University, Nanjing, China, from 1986 to 1991, a Research and Teaching Assistant with Auburn University, from 1991 to 1994, and a Post-Doctoral Research Associate with the University of Maryland, College Park, MD, USA, from 1994 to 1996. He was with AT&T Labs-Research, Red Bank, NJ, USA, as a Senior and a Principal Technical Staff Member, from 1996 to 2000. Since 2000, he has been with the School of Electrical and Computer Engineering, Georgia Institute of Technology, as an Associate Professor and a Full Professor. He has been holding a Cheung Kong Scholar title at the University of Electronic Science and Technology of China since 2006.

His research interests include statistical signal processing and machine learning for wireless communications. In these areas, he has published over 200 journal papers in addition to over 40 granted patents and many conference papers. His publications have been cited over 31,000 times and he has been recognized as the World's Most Influential Scientific Mind, also known as a Highly Cited Researcher, by Thomson Reuters almost every year. He was awarded an IEEE Fellow for his contributions to signal processing for wireless communications in 2005. He received the 2010 IEEE ComSoc Stephen O. Rice Prize Paper Award, the 2013 IEEE VTS James Evans Avant Garde Award, the 2014 IEEE VTS Jack Neubauer Memorial Award, the 2017 IEEE ComSoc Award for Advances in Communication, and the 2017 IEEE SPS Donald G. Fink Overview Paper Award. He also received the 2015 Distinguished Faculty Achievement Award from the School of Electrical and Computer Engineering, Georgia Tech. He has been involved in editorial activities for over 20 technical journals for the IEEE, including the founding Editor-in-Chief of the IEEE 5G Tech Focus. He has organized and chaired many international conferences, including the Technical Program Vice-Chair of the IEEE ICC 2003, the Technical Program Co-Chair of the IEEE SPAWC 2011, the General Chair of the IEEE GlobalSIP 2014, and the Technical Program Co-Chair of the IEEE VTC 2016 (Spring).



H. Vincent Poor (S'72–M'77–SM'82–F'87) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor of electrical engineering. From 2006 to 2016, he served as Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests include information theory and signal processing, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the recent book *Information Theoretic Security and Privacy of Information Systems* (Cambridge University Press, 2017).

Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences and a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He received the Marconi and Armstrong Awards of the IEEE Communications Society in 2007 and 2009, respectively. Recent recognition of his research includes the 2017 IEEE Alexander Graham Bell Medal, Honorary Professorships at Peking University and Tsinghua University, both conferred in 2017, and the D.Sc. *honoris causa* from Syracuse University also in 2017.