Distributed File Allocation Using Matching Game in Mobile Fog-Caching Service Network

Tingting Liu*†, Jun Li†, BaekGyu Kim‡, Chung-Wei Lin‡, Shinichi Shiraishi‡, Jiang Xie§, and Zhu Han¶
*School of Communication Engineering, Nanjing Institute of Technology, Nanjing 211167, CHINA
†School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, CHINA
‡Toyota InfoTechnology Center, USA
§Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC 28223, USA
¶Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA
E-mail: liutt@njit.edu.cn; jleesr80@gmail.com; bkim@us.toyota-itc.com; cwlin@us.toyota-itc.com;
sshiraishi@us.toyota-itc.com; jxie1@uncc.edu; hanzhu22@gmail.com

***Abstract*—The fog-caching enabled radio access network has been identified as an effective solution to reduce content downloading latency of mobile users (MUs). The quality of service (QoS) can be dramatically enhanced by caching popular files into the storages of femto-cell access points (FAPs) adjacent to MUs. However, file allocation is challenging when the mobility of users is considered. In this scenario, the serving users of each FAP are not fixed, and they vary from time to time. In order to address this, we design distributed algorithms, using two separate matching games, with the aim to minimize the system file downloading latency. To be specific, we first allocate files to each FAP and then associate an MU with a proper FAP, considering the mobility of users. At last, numerical results are presented to demonstrate the effectiveness of the proposed algorithms. It is verified that the proposed algorithms outperform the benchmark in terms of achieving a lower transmission delay.**

***Index Terms*—Matching game, mobile fog caching, resource allocation, mobile users**

I. Introduction

Recently, the demands on mobile services with high date rate and low latency are expected to dramatically increase in future wireless networks, especially in the fifth generation (5G) of cellular networks. This is mainly driven by the ever-increasing interest in various multimedia services induced by mobile users (MUs), such as streaming of videos [1]. Against this background, heterogeneous networks, consisting of macro cells, micro cells, and femto cells, have been proven to be capable of meeting this data explosion [2–4]. It is shown that femto-cell access points (FAP), with low costs and low powers, can be deployed in any area underlaid coexisting with various base stations, aiming to improve the overall network performance. Embedding pico/femto/micro base-stations in a macro-cell network is a promising method for achieving substantial gains in coverage and capacity relative to using a macro-cell only. But when FAPs' density in heterogeneous networks increases, backhaul capacity limitations become a non-trivial problem [5].

Wireless content caching is a well-known mechanism to effectively handle backhaul capacity limitations [6]. Network observations demonstrate that a large amount of data traffic is caused by a small portion of popular files, which means that pre-caching some hot files in the storage of network facilities, such as FAPs, will reduce the transmission distance by obtaining data from local nodes rather than from central server nodes. Also, the price of storage medium is relatively lower compared to the price of backhaul. Thus, caching at the edge of network nodes, namely, fog caching, becomes a potential solution recently to relieve backhaul pressures [7–11].

Although different aspects of caching concept in heterogeneous networks are discussed in literatures, while efficient and high performance resource allocation in caching-enabled heterogeneous networks still faces great challenges, especially when MU's mobility is considered. To address so, we utilize the matching game theory to handle resource allocation problems in mobile fog-caching enabled systems. Matching game has been widely used in wireless communications. For instance, the authors in [12] introduced fundamental and conventional classification of matching games for future wireless networks. Generally, matching games are divided into three categories, including one-to-one, many-to-one and many-to-many matching games. In [13], authors utilized many-to-one matching games in wireless small-cell networks with a combination of context-aware of information about trajectory profile and quality of service requirements of users, for maximizing the satisfaction ratio and reduce the downloading delay. In this model, the preferences of the players are interdependent and contingent on the matching structure. They proposed a novel algorithm that converges to a stable matching in a reasonable number of iterations. Many-to-many matching games have been utilized in [14] to reduce backhaul loads and the experienced delay in small-cell networks. In [15], the authors proposed different algorithms using matching games to optimize the total satisfaction of the user equipments in an uplink OFDMA network.

In this paper, we propose a series of matching algorithms based on two-tier matching games to facilitate a caching-oriented resource allocation mechanism in heterogenous networks. We first propose a matching model to allocate files to FAPs, and then associate an MU with proper FAPs. The FAPs and MUs are selfish and rational entities, indicating that

978-1-5386-5979-3/18/$31.00 ©2018 IEEE

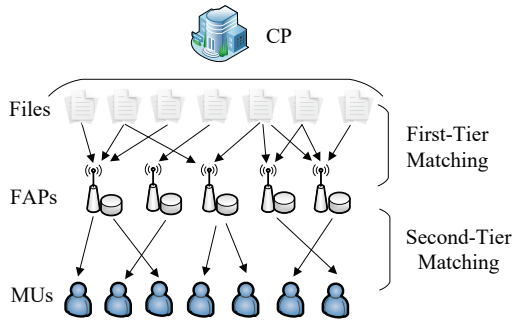

Fig. 1. System architecture.

they only want to maximise their own interests, e.g., minimal transmission latency, lower energy consumption, or higher data rate, etc.

The key contributions of this paper are summarized as follows.

1) We model a distributed mobile caching system containing FAPs, MUs, and one content provider (CP), where the objective is to minimize the system transmission delay.
2) We decouple the resource allocation problem as a two-tier matching game, where in each matching game, the algorithm converges to a high performance and stable matching outcome. The mobility of MUs is considered in the second-tier matching.
3) We demonstrate that the proposed algorithms have near-optimal outcomes with numerical results.

The rest of this paper is organized as follows. We describe the system model in Section II. Matching definition and optimization problems are formulated in Section III. We then propose distributed matching algorithms for dynamic mobile environment in Section IV. The numerical results are illustrated in Section V, and our conclusions are provided in Section VI.

II. System Model

We consider a wireless network which has one CP, several FAPs and multiple MUs. System architecture is shown in Fig 1, where M MUs try to download files with the potential aid of N FAPs. We denote the set of these M MUs by $\boldsymbol{\mathcal{U}} = \{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_M\}$, denote the set consisting of the N FAPs by $\boldsymbol{\mathcal{A}} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_N\}$, and denote the CP by $\mathcal{C}$. We assume that FAPs are randomly distributed and the MUs are randomly moving in the considered network. $\mathcal{C}$ assigns the network contents to N FAPs during off-peak time via backhaul links, for the purpose of decreasing MUs' downloading delay during peak time. We also assume that all the FAPs have the same but constraint storage size, i.e., it can cache limited number of files. Once the files are allocated to FAPs, these FAPs will be able to transmit files directly to the covered MUs upon requests received.

A. File Popularity

We denote the library of the V popular files as $\boldsymbol{\mathcal{F}} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_V\}$. We assume that the file popularity is characterized by the Zipf distribution, where the number of requests to a file is inversely proportional to the file's rank v in the request table [16]. Then, the file popularity q_v of the file $\mathcal{F}_v$ is given by

$$q_v = \frac{\frac{1}{v^\beta}}{\sum_{i=1}^{V} \frac{1}{i^\beta}}, \quad \forall v = 1, \ldots, V, \tag{1}$$

where the exponent β is a positive skewness parameter. Following (1), we note that the file with a smaller index v is of a higher file popularity. Files are assumed to have a uniform length with L bits.

B. Transmission Delay

We denote by $R_{m,n}$ the transmission rate from an FAP $\mathcal{A}_n$, $\forall n = 1, \ldots, N$, to an MU $\mathcal{U}_m$, $\forall m = 1, \ldots, M$. If an MU cannot obtain the requested files from its associated FAPs, it will be redirected to the central servers located at the backbone network, with a transmission rate denoted by R_m. Note that the overhead of redirecting decision is negligible compared to the file downloading. Generally, we have

$$R_m < R_{m,n}. \tag{2}$$

Also, given MU $\mathcal{U}_m$ connects to FAP $\mathcal{A}_n$, the downloading delay of a file from $\mathcal{A}_n$ to $\mathcal{U}_m$ can be written as

$$\tau_{m,n} = \frac{L}{R_{m,n}}. \tag{3}$$

In general, an MU can be covered by multiple FAPs. When MU $\mathcal{U}_m$ requests a file $\mathcal{V}_v$, it will connect to the nearest FAP that caches the requested file. An MU will be redirected to the central server for the files if and only if this MU cannot obtain them from the adjacent FAPs. Also we define the transmission delay from central server to the MU $\mathcal{U}_m$ as

$$\tau_{m,a} = \frac{L}{R_m}. \tag{4}$$

C. Caching Procedure

Next, we introduce the caching procedure with details. In the first stage, the CP intends to rent the storage of the FAPs. We assume that both CP and FAPs are selfish and rational, all competing for their own benefits. CP wants to occupy more FAPs in order to gain more MU customers and offer better services, e.g., lower transmission latency. It is evident that the CP can earn higher profits if it has more affiliated MUs. On the other side, the FAPs are motivated by the monetary payment provided by the CP, and try to gain their profits as much as possible. In this paper, we focus on file allocation and MU association problems in order to reduce the system transmission latency. Incentive mechanism design is not in the scope of this paper. For simplicity, we assume that the FAPs have been properly motivated.

In the second stage, after renting the storage of FAPs, CP will determine the subset of $\boldsymbol{\mathcal{F}}$ allocated to each FAP. Each MU is a distinct individual to express its interests towards files. It is natural to assume that MUs express various interests towards different files. Consequently, we define the preference,

i.e., interests towards files, of the MU $\mathcal{U}_m$ to a specific file $\mathcal{F}_v$ as [17, 18]

$$p_{m,v} = \frac{\alpha_{m,v}}{\alpha_m} q_v, \tag{5}$$

where $\alpha_{m,v}$ is a factor affecting the $\mathcal{U}_m$'s preference to $\mathcal{F}_v$, q_v is the popularity of the file $\mathcal{F}_v$, and $\alpha_m = \sum_v \alpha_{m,v}$ is to normalize $\alpha_{m,v}$.

The probability that an FAP $\mathcal{A}_n$ will cache a file relies on collecting the preferences of its served MUs. Then we define the preference of $\mathcal{A}_n$ to the file $\mathcal{F}_v$, denoted by $w_{n,v}$, as

$$w_{n,v} = \frac{1}{|\boldsymbol{\mathcal{H}}_n|} \sum_{l \in \boldsymbol{\mathcal{H}}_n} p_{l,v}, \tag{6}$$

where l represents the l-th served MU of the FAP, $\boldsymbol{\mathcal{H}}_n$ is the set of the served MUs of $\mathcal{A}_n$, and $|\cdot|$ represents the cardinality of a set. An FAP will cache the most preferred file sets of its serving MUs according to (6). At last, MUs start to contact FAPs for file downloading.

III. Problem Formulations

We now formulate two problems in our caching system. The two problems are linked with each other such that the previous matching results will be transferred to the next matching problem to influence their decision results. In what follows, we will introduce some key concepts to facilitate the problem formulation.

A. Matching Related Definition

In this paper, two matching problems will be tackled. The first matching problem is defined among the FAPs and files for minimizing the system transmission delay. The second game is to decide which FAP an MU should associate with, i.e., the matching between the MUs and the FAPs, to further reduce system transmission delay.

More specifically in the first game, the FAPs and CP's files will be regarded as the two sides of players, and are divided into two finite and disjoint sets. The file allocation procedure will be solved by using a many-to-many matching model, since a file can be cached to multiple FAPs, and each FAP can in turn store multiple distinct files. The strategy of an FAP is to cache relatively popular files, while the strategy at the file (or CP) side is to select a FAP which is capable of offering low latency to its MUs. In a many-to-many matching game, a player in each set has a preference list over members of the opposite set. To construct the preference lists in this model, the symbol $\succ$ is used to represent that a player prefers one player over another player in the opposite set. For example, when a FAP $\mathcal{A}_n$ shows $\mathcal{F}_1 \succ \mathcal{F}_2$ in its preference lists, it means that $\mathcal{A}_n$ prefers the file $\mathcal{F}_1$ over file $\mathcal{F}_2$.

In this paper, we define μ_1 as the matching process of the first matching game. To be specific, for $\mathcal{F}_v \in \boldsymbol{\mathcal{F}}$ and $\mathcal{A}_n \in \boldsymbol{\mathcal{A}}$, a matching μ_1 is represented by $\boldsymbol{\mathcal{F}} \bigcup \boldsymbol{\mathcal{A}} \rightarrow 2^{\boldsymbol{\mathcal{F}} \cup \boldsymbol{\mathcal{A}}}$, which satisfies the following,

1) $\mu_1(\mathcal{A}_n) \subset \boldsymbol{\mathcal{F}}$ and $|\mu_1(\mathcal{A}_n)| \leq Q_{\boldsymbol{\mathcal{F}}}$,
2) $\mu_1(\mathcal{F}_v) \subset \boldsymbol{\mathcal{A}}$ and $|\mu_1(\mathcal{F}_v)| \leq Q_{\boldsymbol{\mathcal{A}}}$,
3) $\mu_1(\mathcal{F}_v) = \mathcal{A}_n \Longleftrightarrow \mu_1(\mathcal{A}_n) = \mathcal{F}_v$,

where item 1) represents that the matching players of $\mathcal{A}_n$ are contained in $\boldsymbol{\mathcal{F}}$, and each FAP can cache at most $Q_{\boldsymbol{\mathcal{F}}}$ files, with $Q_{\boldsymbol{\mathcal{F}}}$ being the maximum number of files that an FAP can cache, item 2) means that the matching players of $\mathcal{F}_v$ are contained in $\boldsymbol{\mathcal{A}}$, and each file can be cached by at most $Q_{\boldsymbol{\mathcal{A}}}$ FAPs, with $Q_{\boldsymbol{\mathcal{A}}}$ denoting the maximum number of the FAPs that a file can be copied in, considering the copyright constraint, and item 3) implies that if $\mathcal{F}_v$ is matched to $\mathcal{A}_n$, then $\mathcal{A}_n$ is matched to $\mathcal{F}_v$, and vice versa.

In the second matching game, i.e., MU allocation matching problem, the game players are the FAPs and MUs. Each FAP's strategy is to decide whether to accept or reject requests from the MUs, while each MU wants to associate with the best FAP based on the outcome of the first-tier matching. Since an FAP can possibly cover multiple MUs and an MU is allowed to connect to one FAP, this matching game is processed as a many-to-one matching with the following properties.

For $\mathcal{U}_m \in \boldsymbol{\mathcal{U}}$ and $\mathcal{A}_n \in \boldsymbol{\mathcal{A}}$, a matching μ_2 is defined by $\boldsymbol{\mathcal{U}} \bigcup \boldsymbol{\mathcal{A}} \rightarrow 2^{\boldsymbol{\mathcal{U}} \cup \boldsymbol{\mathcal{A}}}$, which satisfies

1) $\mu_2(\mathcal{U}_m) \in \boldsymbol{\mathcal{A}}$ and $|\mu_2(\mathcal{U}_m)| \leq 1$,
2) $\mu_2(\mathcal{A}_n) \subset \boldsymbol{\mathcal{U}}$ and $|\mu_2(\mathcal{A}_n)| \leq Q_{\boldsymbol{\mathcal{U}}}$,
3) $\mu_2(\mathcal{U}_m) = \mathcal{A}_n \Longleftrightarrow \mu_2(\mathcal{A}_n) = \mathcal{U}_m$,

where item 1) means that one MU can only be associated with one FAP, item 2) represents that a FAP could serve at most $Q_{\boldsymbol{\mathcal{U}}}$ MUs, and item 3) states that if $\mathcal{A}_n$ is matched to $\mathcal{U}_m$, then $\mathcal{U}_m$ is matched to $\mathcal{A}_n$, and vice versa.

B. First Matching: Files-FAPs Matching

In this subsection, we formulate the files-FAPs matching problem to minimize the transmission delay of MUs. The problem can be formulated as:

$$\min_{\boldsymbol{X}} \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{v=1}^{V} \left(x_{n,v}\tau_{m,n} + (Q_{\boldsymbol{\mathcal{F}}} - \sum_{v} x_{n,v})\tau_{m,a} \right) w_{n,v}, \tag{7}$$

$$\text{s.t.}(a) \sum_{v} x_{n,v} \leq Q_{\boldsymbol{\mathcal{F}}},\ (b) \sum_{n} x_{n,v} \leq Q_{\boldsymbol{\mathcal{A}}},\ (c)\ x_{n,v} \in \{0,1\},$$

where $x_{n,v}$ is the element of the matrix $\boldsymbol{X}$. $x_{n,v} = 1$ represents that the FAP $\mathcal{A}_n$ caches the file $\mathcal{F}_v$, while $x_{n,v} = 0$, otherwise. Condition (a) guarantees that each FAP can cache at most $Q_{\boldsymbol{\mathcal{F}}}$ files. Concerning file copyright issue, we set the condition (b) to make sure that the file $\mathcal{F}_v$ can only be cached with $Q_{\boldsymbol{\mathcal{A}}}$ duplications in this network, and condition (c) implies that the values of $x_{n,v}$ can only be a binary value 0 or 1. The optimization in (7) is an NP-hard combinatorial problem [19], which has a high computational complexity.

C. Second Matching: MUs-FAPs Matching

After the first matching is determined, the MU allocation problem can be modeled as follows,

$$\min_{\boldsymbol{Y}} \sum_{n=1}^{N} \sum_{m=1}^{M} y_{n,m}\tau_{m,n} + (1 - \sum_{n} y_{n,m})\tau_{m,a}, \tag{8}$$

$$\text{s.t.}(a) \sum_{n} y_{n,m} \leq 1,\ (b) \sum_{m} y_{n,m} \leq Q_{\boldsymbol{\mathcal{U}}},\ (c)\ y_{n,m} \in \{0,1\},$$

where $y_{n,m}$ is the element in the matrix $\boldsymbol{Y}$. $y_{n,m} = 1$ represents that FAP $\mathcal{A}_n$ serves MU $\mathcal{U}_m$, and otherwise, $y_{n,m} = 0$. Condition (a) states that a user will be served by only one FAP. Condition (b) ensures that each FAP can serve at most $Q_{\boldsymbol{\mathcal{U}}}$ MUs, and condition (c) states that the values of $y_{n,m}$ can be either 0 or 1. It is again an NP-hard combinatorial problem.

IV. Distributed Matching Algorithms for Dynamic Mobile Environment

In this section, we propose two different matching algorithms to solve the above problems. The proposed algorithms are all based on matching theory. Deferred Acceptance (DA) algorithm [20] is proposed to solve the first matching. For the second matching, considering the mobility of MUs, Roth Vanda-Vate (RVV) algorithm is proposed to solve the dynamic matching problem [21]. Those distributed matching algorithms fit the dynamic mobile environment well.

A. Files-FAPs Matching Algorithm

To process file selection algorithm, we define FAPs' preference lists firstly. The cached files set of each FAP is a key factor that influence the serving MUs' choices, so we choose the serving MUs' preferences over files to build FAPs' preference lists. The definition of FAP's preference lists is shown as follows.

Definition 1: FAP's preference over file $\mathcal{F}_v \in \boldsymbol{\mathcal{F}}$ is defined as

$$\boldsymbol{\Gamma}^{\text{FAP}} = w_{n,v}, \tag{9}$$

where $\boldsymbol{\Gamma}^{\text{FAP}} \in \mathbb{C}^{N \times V}$ is FAPs' preference matrix over files.

Also, files have certain preferences towards different FAPs considering transmission delay. A FAP can serve $\mathcal{H}_n$ MUs so that we take the average transmission delay of $\mathcal{H}_n$ serving MUs as file's preference over this FAP. We define it as follows.

Definition 2: File's preference over FAP $\mathcal{A}_n \in \boldsymbol{\mathcal{U}}$ can be given as

$$\boldsymbol{\Gamma}^{\text{CP}} = \frac{1}{|\boldsymbol{\mathcal{H}_n}|} \sum_{l \in \mathcal{H}_n} \tau_{l,n}, \tag{10}$$

where $\boldsymbol{\Gamma}^{\text{CP}} \in \mathbb{C}^{V \times N}$ is the files' preference matrix over FAPs.

As shown in **Algorithm 1**, we use $\boldsymbol{\Gamma}^{\text{FAP}}$ and $\boldsymbol{\Gamma}^{\text{CP}}$ as preference lists. We assume that the number of files is larger than the number of FAPs. At first, the FAPs send their offers to files according to their preference list. Since a file can be cached in $Q_{\boldsymbol{\mathcal{A}}}$ FAPs, we need to judge whether the requests from FAPs for a file are more than the quota $Q_{\boldsymbol{\mathcal{A}}}$. If there are more than $Q_{\boldsymbol{\mathcal{A}}}$ FAPs requesting the same file, files will select its most preferred $Q_{\boldsymbol{\mathcal{A}}}$ FAPs according to its preference list. If the requests for a file are less than or equal to the quota $Q_{\boldsymbol{\mathcal{A}}}$, files will accept all these requests. The accepted FAPs should judge whether they have free memory space for other files. If the FAP has, it will be still in unmatched set $\boldsymbol{\mathcal{R}}_1$, otherwise it will be removed from unmatched list. The unmatched FAPs will continue to take part in the next round of proposals until each FAP caches $Q_{\boldsymbol{\mathcal{F}}}$ files. We will get a stable matching μ_1 in the end.

TABLE I
PROPOSED FIRST MATCHING ALGORITHM

Algorithm 1 Files-FAPs Matching Algorithm

Input: $\boldsymbol{\Gamma}^{\text{FAP}}$, $\boldsymbol{\Gamma}^{\text{CP}}$, $Q_{\boldsymbol{\mathcal{F}}}$, $Q_{\boldsymbol{\mathcal{A}}}$;
Output: stable matching μ_1;
Steps:
construct empty space for unmatched FAPs to files as sets of $\boldsymbol{\mathcal{R}}_1 = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n\}$.
while $\boldsymbol{\mathcal{R}}_1 \neq \emptyset$ **do**
  for all $\mathcal{A}_n \in \boldsymbol{\mathcal{R}}_1$ do
    sends offer to first file in its preference list according to $\boldsymbol{\Gamma}^{\text{FAP}}$ and set $x_{n,v} = 1$;
    remove the first choice in $\boldsymbol{\Gamma}^{\text{FAP}}$;
  end for
  for all $\mathcal{F}_v \in \boldsymbol{\mathcal{F}}$ **do**
    if $\sum_n x_{n,v} > Q_{\boldsymbol{\mathcal{A}}}$
    the file choose most preferred FAPs according to $\boldsymbol{\Gamma}^{\text{CP}}$ and the other FAPs will be rejected;
    the rejected FAPs will be set $x_{n,v} = 0$;
    else
    the offers from FAPs will all be accepted;
    end if
  end for
  for all $\mathcal{A}_n \in \boldsymbol{\mathcal{R}}_1$ **do**
    if $\sum_v x_{n,v} > Q_{\boldsymbol{\mathcal{F}}}$
    remove $\mathcal{A}_n$ whose index is larger than $Q_{\boldsymbol{\mathcal{F}}}$ from $\boldsymbol{\mathcal{R}}_1$;
  end for
end while
return stable matching μ_1.

B. MUs-FAPs Matching Algorithm

After solving file allocation problem using **Algorithm 1**, we turn to handle the MU association problem. First we define the preference of FAP over MUs as follows.

Definition 3: For a FAP $\mathcal{A}_n \in \boldsymbol{\mathcal{A}}$, its preference over MU $\mathcal{U}_m \in \boldsymbol{\mathcal{U}}$ can be given as

$$\boldsymbol{\Gamma}^{\text{FM}} = \tau_{m,n}, \tag{11}$$

where $\boldsymbol{\Gamma}^{\text{FM}} \in \mathbb{C}^{M \times N}$ is the FAPs' preference matrix over MUs.

Moreover, the preference of MUs over FAPs will be affected by the first matching result. After the first matching, FAPs will broadcast information about the cached files to all its nearby MUs. Since different MUs have different preferences over files and thus they have different preferences over FAPs. Then, we define MU's preference over FAPs as follows.

Definition 4: For $\mathcal{U}_m \in \boldsymbol{\mathcal{U}}$, its preference over FAP $\mathcal{A}_n \in \boldsymbol{\mathcal{A}}$ can be given as

$$\boldsymbol{\Gamma}^{\text{MTF}} = \sum_{v=1}^{V} \tau_{m,n} p_{m,v} x_{n,v}, \tag{12}$$

where $\boldsymbol{\Gamma}^{\text{MTF}} \in \mathbb{C}^{M \times N}$ is the MUs' preference matrix over FAPs.

The fixed MU's association problem can be solved by **Algorithm 2**. Considering MUs' mobility, we follow the same analysis steps in [22] that a time slot is divided into multiple small time slots $\triangle T$ such that within $\triangle T$, the MU can be treated as static. The specific algorithm of second matching is shown in **Algorithm 3**. The time-dependent algorithm based on RVV has been proposed to minimize average transmission delay of MUs by associating MUs to FAPs. As shown in

TABLE II
PROPOSED SECOND MATCHING ALGORITHM - PART 1

Algorithm 2 MUs-FAPs Matching Algorithm

Input: $\mathbf{\Gamma}^{\mathrm{FM}}$, $\mathbf{\Gamma}^{\mathrm{MTF}}$, $Q_{\mathcal{U}}$;
Output: stable matching μ_2;
Steps:
construct sets of unmatched MUs to FAPs as $\boldsymbol{\mathcal{R}}_2 = \{\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_m\}$.
while $\boldsymbol{\mathcal{R}}_2 \neq \emptyset$ and $\sum y_{n,m} < Q_{\mathcal{U}}$ **do**
  for all $\mathcal{U}_m \in \boldsymbol{\mathcal{R}}_2$ do
    sends offer to first FAP in its preference list due to $\mathbf{\Gamma}^{\mathrm{MTF}}$ and set $y_{n,m} = 1$;
    remove the first choice in $\mathbf{\Gamma}^{\mathrm{MTF}}$;
  end for
  for all $\mathcal{A}_n \in \boldsymbol{\mathcal{A}}$ **do**
    if $\sum_m y_{n,m} > Q_{\mathcal{U}}$
    the file choose most preferred MUs due to $\mathbf{\Gamma}^{\mathrm{FM}}$ and the other MUs will be rejected;
    the rejected MUs set $y_{n,m} = 0$;
    else
    the offers from MUs will all be accepted;
    end if
  end for
end while
the unmatched MUs will have a connection with central server;
return stable matching μ_2.

TABLE III
PROPOSED SECOND MATCHING ALGORITHM - PART 2

Algorithm 3 MUs with Mobility Matching Algorithm

Input: stable matching $\mu_2(t-1)$ in the previous time t-1;
Output: stable matching $\mu_2(t)$ at time t;
Steps:
Initialization:
$\mu_2 = \mu_2(t-1)$, $I = \oslash$;
while μ_2 is not stable **do**
  if There exists$(\mathcal{U}_m, \mathcal{A}_n) \in bp(\mu_2)$ such that $\mathcal{U}_m \notin I$ and $\mathcal{A}_n \in I$, **then add** $\mathcal{U}_m$;
  else
    choose $(\mathcal{U}_m, \mathcal{A}_n) \in BP(\mu_2)$;
    satisfy $(\mathcal{U}_m, \mathcal{A}_n)$;
  end if
end while
$\mu_2 = \mu_2(t)$;

Algorithm 3, the algorithm starts from an initial matching μ_2, which is a stable matching obtained from **Algorithm 2**. Moreover, it is $\mu_2(t-1)$ from the previous time slot $t-1$. A set I is utilized during the iterations of the algorithm, which is initially empty. The algorithm iterates as long as μ_2 is not stable. During each iteration, if there is a BP $(\mathcal{U}_m, \mathcal{A}_n)$ such that $\mathcal{U}_m \notin I$ and $\mathcal{A}_n \in I$, the procedure '**add**' is called with $\mathcal{U}_m$. Otherwise, the '**satisfy**' procedure is called with $\mathcal{U}_m$ and $\mathcal{A}_n$, i.e., $\mathcal{U}_m \notin I$ and $\mathcal{A}_n \notin I$.

C. Overhead and Complexity

Matching algorithms are implemented in a distributed manner, so it will cause less overhead and complexity compared to the centralized algorithm. The number of communication packets of the algorithms is hard to analyze because the system parameters are set independently and they have mutual interference. However, we can analyze the upper bound of communication packets for the first matching algorithm. Considering the time scale of the proposed algorithm, until the algorithm converges, the signaling packet length required for the communication between the players is very short.

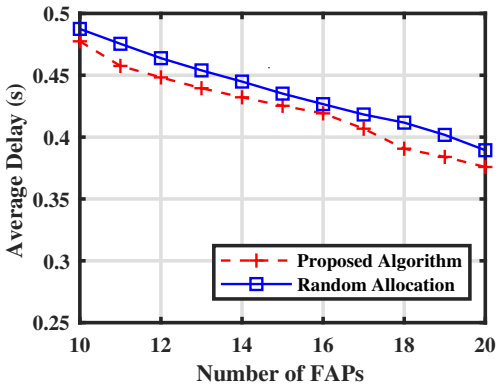

Fig. 2. Average delay vs. the number of FAPs in first matching.

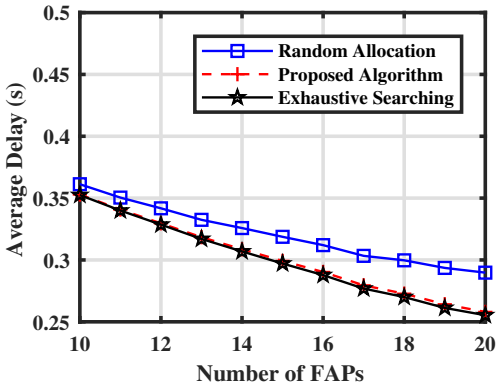

Fig. 3. Average delay vs. the number of FAPs in second matching.

Theorem 1: The number of communication packets between the files and the FAPs required in the first matching is upper bounded by

$$\mathfrak{N}_{max} = Q_{\boldsymbol{\mathcal{A}}} N(N-1)(N-1). \tag{13}$$

Proof: For clarity, we omit the proof here. ■

Theorem 2: A stable matching μ_2 can be obtained in $O\left((N+M)\,m\right)$ times, where m is the number of acceptable MU-FAP pairs in I.

Proof: For clarity, we omit the proof here. ■

V. Numerical Results

In this section, we evaluate the performance of the proposed two algorithms by numerical results using MATLAB. We assume $V = 50$ and $Q_{\boldsymbol{\mathcal{A}}} = 2$. In this simulation, we assume FAPs are randomly located, and MUs are moving under Random Waypoint Model (RWP) in the network [23]. Each node in the RWP model begins by stopping for a fixed time, and then moves to a random direction with a random speed between $[0, v_o]$, in which v_o represents the maximum speed. This movement pattern repeats until the end of the simulation.

In the following figures, we compare our proposed algorithms with random allocation and exhaustive searching algorithms. In the random allocation algorithm, files are randomly cached. In the exhaustive searching algorithm, the problems are solved by the centralized solution with high complexity.

In Fig. 2, the number of files is 50 and the number of FAPs varies from 10 to 20. Each FAP can cache $Q_{\boldsymbol{\mathcal{F}}} = 2$ files at

most. As shown in Fig. 2, with the increasing of the number of FAPs, the average delay of both the random allocation algorithm curve and proposed algorithm curve have a declining trend. And the proposed algorithm shows a better performance than random allocation scheme.

In Fig. 3, the number of MUs is 60 and the number of FAPs varies from 10 to 20 with each FAP serving at most $Q_{\mathcal{U}} = 2$ MUs. As observed from Fig. 3, with the increasing of the number of FAPs, the average delay of both the exhaustive searching algorithm curve and proposed algorithm curve have a decreasing trend. Though the exhaustive searching algorithm shows better performance to the proposed one, the proposed algorithm has less computation complexity while the exhaustive searching algorithm complexity increases exponentially over network size. Obviously, with low complexity, the proposed algorithm will reduce system processing power. And we find that the random allocation algorithm exhibits inferior performance compared to the proposed algorithm.

VI. Conclusion

In this paper, the file allocation problems in a fog caching network was studied. The caching problems were modeled as two-tier matching games which include the first matching between files and FAPs, and the second matching between FAPs and MUs with mobility. Two different matching algorithms are proposed to solve the two-tier matching problem. The upper bound of communication packets for the first-tier matching algorithm and the computational complexity of the secondary-tier matching are also provided. Finally, simulation results are provided to demonstrate that the proposed algorithms have highly comparable performance with the exhaustive searching algorithm in reducing average transmission delay and have better performance than the random allocation algorithm.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grants 61702258, 61771244, 61727802, 61501238 and 61472190, in part by the Jiangsu Provincial Science Foundation under Project BK20150786, in part by the Specially Appointed Professor Program in Jiangsu Province, 2015, in part by the Fundamental Research Funds for the Central Universities under Grant 30916011205, and in part by the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University, under grant No. 2017D04 and 2013D02, in part by the China Postdoctoral Science Foundation under grant 2016M591852, in part by Postdoctoral research funding program of Jiangsu Province under grant 1601257C, in part by the China Scholarship Council Grant 201708320001, partially supported by US NSF CNS-1717454, CNS-1731424, CNS-1702850, CNS-1718666 CNS-1731675, CNS-1646607, ECCS-1547201.

References

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

[2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[3] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Communications*, vol. 18, no. 3, pp. 10–21, Jun. 2011.

[4] J. Akhtman and L. Hanzo, "Heterogeneous Networking: An Enabling Paradigm for Ubiquitous Wireless Communications [Point of View]," *IEEE Journal on Proceedings*, vol. 98, no. 2, pp. 135–138, Feb. 2010.

[5] T. Q. Quek, *Small cell networks: Deployment, PHY techniques, and resource management*. Cambridge University Press, 2013.

[6] 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 Communications Magazine*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[7] 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 Transactions on Communications*, vol. 63, no. 10, pp. 3553–3568, Oct. 2015.

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

[9] L. Wang, H. Wu, Y. Ding, W. Chen, and H. V. Poor, "Hypergraph-Based Wireless Distributed Storage Optimization for Cellular D2D Underlays," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2650–2666, Oct. 2016.

[10] B. Bai, L. Wang, Z. Han, W. Chen, and T. Svensson, "Caching based socially-aware D2D communications in wireless content delivery networks: a hypergraph framework," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 74–81, Aug. 2016.

[11] T. Liu, J. Li, F. Shu, M. Tao, W. Chen, and Z. Han, "Design of contract-based trading mechanism for a small-cell caching system," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6602–6617, Oct. 2017.

[12] S. Bayat, Y. Li, L. Song, and Z. Han, "Matching theory: Applications in wireless communications," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 103–122, Nov. 2016.

[13] N. Namvar, W. Saad, B. Maham, and S. Valentin, "A context-aware matching game for user association in wireless small cell networks," in *Proc. IEEE ICASSP, Florence, Italy*, May 2014, pp. 439–443.

[14] K. Hamidouche, W. Saad, and M. Debbah, "Many-to-many matching games for proactive social-caching in wireless small cell networks," in *Proc. IEEE 12th International Symposium on WiOpt, Hammamet, Tunisia*, May 2014, pp. 569–574.

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

[16] 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 Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2115–2129, Aug. 2016.

[17] Y. Gu, Y. Zhang, M. Pan, and Z. Han, "Student admission matching based content-cache allocation," in *Proc. IEEE Wireless Communications and Networking Conference, WCNC, Istanbul, Turkey*, Mar. 2015, pp. 2179–2184.

[18] Z. Chang, Y. Gu, Z. Han, X. Chen, and T. Ristaniemi, "Context-aware data caching for 5G heterogeneous small cells networks," in *Proc. IEEE International Conference on Communications, ICC, Kuala Lumpur, Malaysia*, May 2016.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[20] Sotomayor and M. A. Oliveira, *Two-sided matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, 1992.

[21] A. E. Roth and J. H. V. Vate, "Random paths to stability in two-sided matching," *Econometrica: Journal of the Econometric Society*, vol. 58, no. 6, pp. 1475–1480, Nov. 1990.

[22] Y. Gu, C. Jiang, L. X. Cai, M. Pan, L. Song, and Z. Han, "Dynamic path to stability in lte-unlicensed with user mobility: A matching framework," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4547–4561, Jul. 2017.

[23] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1998, pp. 85–97.