D2D Based Caching Content Placement in Wireless Cache-Enabled Networks

Junyue Qu, Dan Wu, Yanshan Long, Wendong Yang, Yueming Cai

College of Communications Engineering, Army Engineering University of PLA, China
qujunyue_0610@sina.cn, wujing1958725@126.com, yslong_own@163.com, ywd1110@163.com, caiym@vip.sina.com

Abstract

In order to provide a satisfying content download service and reduce the pressure of the base station, content caching based on the D2D communications is drawing more and more attentions, which allows two requesters in close proximity to share the contents of common interest directly. However, since the storage capacity and the communication ability of the mobile nodes are limited, it is impossible for a mobile node to cache all the files and share the files with any node. Hence, one of the most important issues is the caching content placement for the mobile nodes where the files can be cached. In our work, to achieve the potential advantages of the D2D-based content caching, we consider to maximize the file download rate of the whole network by proper caching content placement. It is proved that the problem of the caching content placement is NP hard. In view of its intractability, we prove that the problem of the caching content placement can be modeled as the maximization of a monotone submodular function over one matroid and multiple knapsack constraints. Accordingly, the greedy algorithm can be utilized to obtain a suboptimal solution. It is proved that the complexity of the proposed caching content placement algorithm based on greedy algorithm is polynomial, and it yields a constant-factor approximation to the problem of the caching content placement. The simulation results show that the nodes which can cache files can perform more efficiently with our proposed caching content placement algorithm based on greedy algorithm.

Keywords: Device-to-device communication, Caching content placement, Submodular function, Knapsack constraint, Greedy algorithm

1 Introduction

With the wireless data traffic increasing dramatically, the content request is predicted to be the main data traffic in the future [1]. Content sharing based on D2D communication emerging as a technology which can take use of the advantage of D2D communications more is being warmly discussed in various aspects [2-6]. It allows two requesters in close proximity to share the contents of common interest directly. In such a way, requesters can enjoy higher data rate and less delay, and the pressure of the BS can be reduced [7-9]. Specially, when a group of requesters are interested in the same content, a mobile node called helper where the files are cached can serve these requesters quickly and the advantages to cache popular contents in helpers are especially obvious. However, to achieve caching content placement based on D2D communication, many challenges need be solved [10]. As such, the content caching has attracted so many attentions [11-16]. However, many problems are introduced with the proposal of the content caching. Firstly, it is impossible for a cache node to cache all the contents due to the limited storage capacity. The storage capacity will be wasted if a file is cached for too many times while the cache hit rate will be low if a file just is cached for one time. Secondly, as the statement in [9], if every requester can just access one helper, it is clear that each helper should cache the most popular ones. However, for the case that every requester can access multiple helpers, the caching content placement of helpers becomes nontrivial. Thirdly, for each helper, the communication ability is limited. A helper can just serve several requesters and even one requester. Obviously, the two cases in [9] are existing at the same time in a wireless cache-enabled network. When we take some other considerations in account, the caching content placement of helpers becomes more complex. As a result, one of the most important issues is the caching content placement.

Up to now, many works have been done to discuss how to accurately place the caching content. In [11], authors consider to design an interference-aware collaborative caching mechanism to make caching decision based on the content popularity. However, it results in that the cooperative caching is achieved to some probability. To enhance the cooperative caching, maybe some constraints else should be considered. The caching content placement is considered from the perspective of the space to maximize the hit probability

*Corresponding Author: Junyue Qu; E-mail: qujunyue_0610@sina.cn
DOI: 10.3966/160792642019032002002

in [12]. However, this work does not consider the channel difference between different cache-enabled users. In [13], authors minimize the energy consumption for the whole network with a suitable caching scheme with a constraint that requesters obtain acceptable rate. However, due to the constraint, requesters cannot obtain higher rate and enjoy better service. In our opinion, for the whole network, it is very important to improve the service of consumers. In [14-15], the method of segment-based caching content placement is utilized. It is helpful to reduce the complexity to obtain the final caching content placement scheme. However, comparing with the method to consider each file one by one, maybe it is an obstacle to reach the optimal caching content placement with the consideration that the files within the same segment have the same cached probability. In [16], a caching content placement scheme for distributed caching helpers is discussed to minimize the expected total file download delay so that the effect of the helpers can be much more improved. However, it is possible that the quality of service (QoS) of some requester is ignored when just the idle and best links of each requester are considered. For example, for a requester who has a better link and multiple worse links, if the better link is ignored, it is possible that the QoS of this requester cannot be met. Maybe it is a good consideration to take all links of one requester in account.

Motivated by the above analysis, we consider the problem of caching content placement based on D2D communications to maximize the whole file download rate of the network for a given file popularity distribution, storage capacity of helper and network topology. The contributions of this paper are shown as follows:

- The problem of caching content placement is modeled as an integer programming problem to maximize the whole file download rate of the requesters. In particular, we design a constraint that a file cannot be cached in different helpers which are adjacent to one requester at the same time so as to achieve the cooperative caching among helpers.
- The problem of caching content placement is proved to be NP hard. In view of its intractability, we prove that the resulting problem can be formulated as the maximization of a monotone submodular set function subject to a matroid constraint and multiple knapsack constraints.
- A caching content placement algorithm based on greedy algorithm is designed to find a suboptimal solution according to the characteristic of our model. We analyze that the complexity of the proposed algorithm is polynomial, and we prove that it yields a constant-factor approximation to the original problem.

The rest of the manuscript is organized as follows. In Section 2, we describe the system model and the problem of caching content placement. We express the problem as a monotone submodular function over one matroid and multiple knapsack constraints in Section 3. Followed, the problem of caching content placement is proved to be NP hard. In Section 4, the caching content placement algorithm based on greedy algorithm is proposed. Some simulations are done in Section 5 to verify the performance of the proposed caching content placement algorithm based on greedy algorithm. Finally, in Section 6, conclusions are drawn.

2 System Model and Problem Formulation

As shown in Figure 1, we consider a network, which consists of a population of K requesters, denoted by $\mathcal{U}=\{u_1,u_2,\cdots,u_K\}$. These requesters are modeled as some content requesters. Besides, there are N helpers where some files can be cached, denoted as $\mathcal{H}=\{h_1,h_2,\cdots,h_N\}$. These helpers work as content providers. Moreover, we assume a set of files which may be cached in the helpers, denoted as $\mathcal{F}=\{f_1,f_2,\cdots,f_F\}$. The files are sorted according to the rank of popularity. The file with the highest rank is the most popular file f_1 . The requesters can request to download the desired files from the helpers instead of the BS. Note that, although storage capacity of the mobile requesters is continuously increasing by a large margin, it is a fact that the storage capacity is very limited comparing to the total number of files. As a result, the helpers cannot cache and provide the contents freely. In addition, a requester can download the desired files from a helper just when the requester is located in the limited communication range of the very helper. Specifically, for requester u_k , the set of the helpers who have the channel to the requester which are good enough to establish D2D links is denoted as $H(u_k)$, $H(u_k)=\{h_i | r_{ki} \geq R_{th}\}$, where r_{ki} is the capacity of the link between the requester u_k and the helper h_i which utilize the model in [17], and R_{th} is the constraint of minimal channel capacity for each requester. It is obvious that a proper caching content placement scheme is very important due to the limited storage capacity and the limited communication range of the helpers.

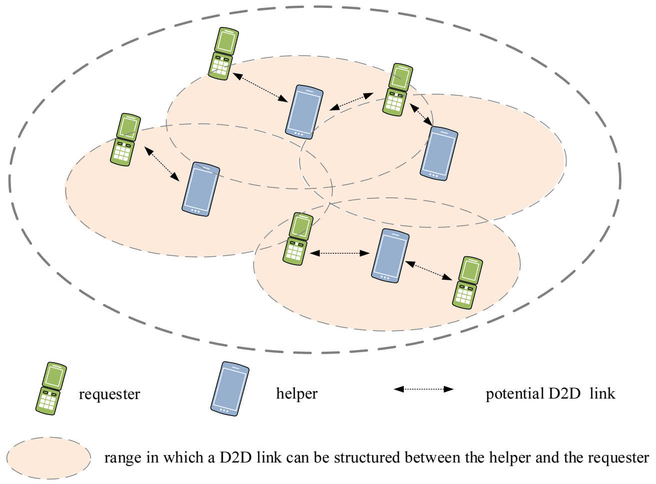


Figure 1. The system model

Due to the limited storage capacity and the limited communication ability of helpers, it is impossible to meet all the request of requesters. In order to improve the efficiency of the helpers, we want to do our best to increase the cache hit rate. The cache hit rate of h_i, can be defined as

$$CHR_i = \sum_{j=1}^{F} x_{ji} P_j \quad \textbf{(1)}$$

where $x_{ji} \in \{0,1\}$ is an indicator whether file f_j is cached in the helper h_i. $x_{ji} = 1$ means that file f_j is cached by helper h_i and $x_{ji} = 0$ means that file f_j is not cached by helper h_i. Besides, P_j is the requesting probability for file f_j.

Considering that most requesters are just interested in a few of contents, we assume P_j obeys the Zipf distribution [16, 20-22]. It can be shown as

$$P_j = \frac{\left(1/j\right)^s}{\sum_{j=1}^{F}\left(1/j\right)^s} \quad \textbf{(2)}$$

where s is the demand dominance factor. Obviously, a too large s will result in a case where just a very small number of contents will be requested. As a result, it is a waste of storage space to discuss the other contents. That is, it is unnecessary to discuss the problem of the caching content placement when s is too large.

In our work, the most important is to provide the best service. We consider that the file download rate is an important and intuitional norm reflecting the QoS of every requester. However, the requests from a requester cannot be known in advance. As a result, we denote the file download rate of the requester u_k as FDR_k which is the function of P_j. Specifically, FDR_k is shown as follows.

$$FDR_k = \sum_{i:h_i \in H(u_k)} \sum_{j=1}^{F} r_{ki} x_{ji} P_j \quad \textbf{(3)}$$

In addition, some practical constraints must be met.For example, the storage capacity of helpers is limited. We assume that the storage capacity of each helper is M. That is, the following inequality must be met.

$$\sum_{j=1}^{F} x_{ji} \le M \quad \textbf{(4)}$$

Also, to make full use of the storage capacity of helpers, for arbitrary $k = 1, \cdots, K$ and $j = 1, \cdots, F$, we set a constraint that different helpers which belongs to $H(u_k)$ cannot cache the same file f_j at the same time. With such a constraint, cooperative caching among helpers can be achieved to meet the comprehensive requests for files of requesters. As shown in Figure 2, two cases are shown to compare with each other. In two cases, requester 1 can obtain the desired files from three helpers. In case a, we consider the proposed constraint above. The three helpers cache different files. Requester 1 can obtain any one of the six files. In case b, we consider that all the three helpers cache file 1 and file 2. Obviously, requester 1 can just obtain file 1 and file 2. Comparing the two cases, the requester can obtain more files in case a. To sum up, the proposed constraint above is very valuable. Such a constraint can be shown as

$$\sum_{i:h_i \in H(u_k)} x_{ji} \le 1 \quad \textbf{(5)}$$

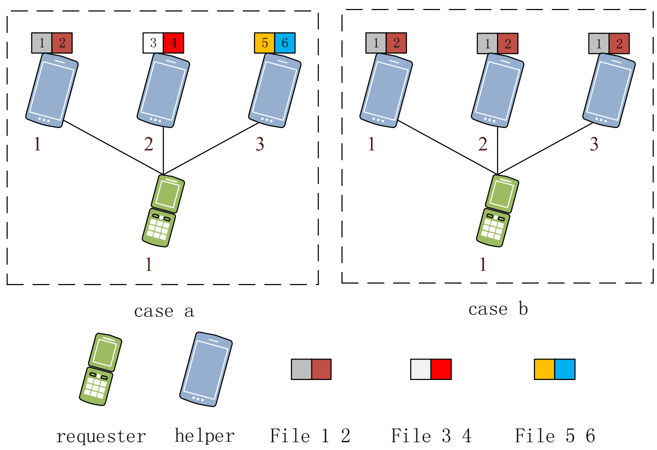
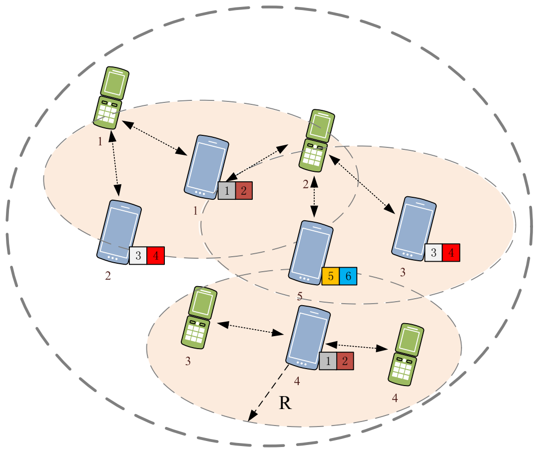
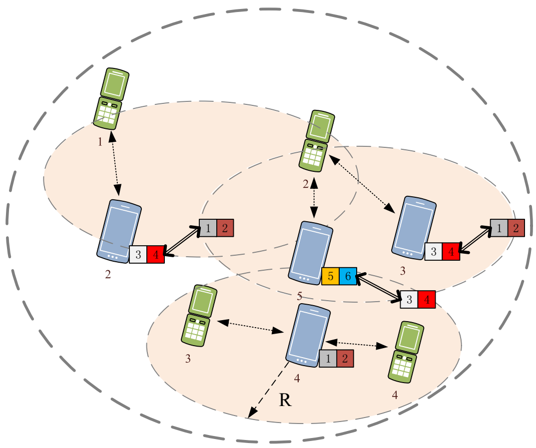


Figure 2. The compare between different caching content placement schemes

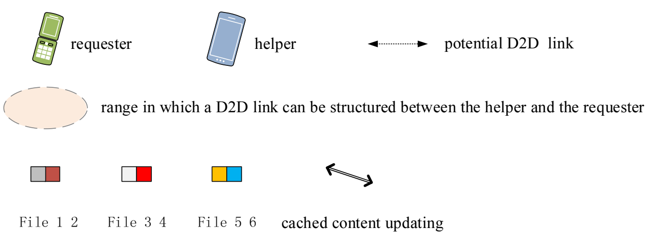
This constraint can achieve the goal of cooperative caching so that the number of files a requester can reach is the largest when we ignore the difference size of two files. It is valuable to note that due to this constraint, when a content helper leaves, especially a content helper where the files which are very popular are cached leaves, the caching content placement will be seriously affected. The details are shown in the Figure 3.



(a) The caching content placement in Scenario 1



(b) The caching content placement in Scenario 2



(c) The legend

Figure 3. The impact of helper's departure on the caching content placement

In Scenario 1, helper 1 caches file 1 and file 2 to serve requester 1 and requester 2. Helper 2 caches file 3 and file 4 to serve requester 1. Helper 3 caches file 3 and file 4 to serve requester 2. Helper 5 caches file 5 and file 6 to serve requester 2. Helper 4 caches file 1 and file 2 to serve requester 3 and requester 4. In the following, we discuss the impact of one helper's departure from two aspects.

In one hand, when helper 1 leaves the network, Scenario 1 is transformed into Scenario 2. Then, helper 2 caches file 1 and file 2 instead of file 3 and file 4 to serve requester 1. Helper 3 caches file 1 and file 2 instead of file 3 and file 4 to serve requester 2. Simultaneously, helper 5 caches file 3 and file 4 instead of file 5 and file 6 to serve requester 2. In addition, Helper 4 still caches file 1 and file 2. Obviously, we can draw two conclusions. Firstly, the departure of one content helper can affect the previous caching content placement. Secondly, the impact of one helper's departure is local. The departure of one content helper affects the helpers who serve the same requesters. In addition, the helpers who serve the same requesters with the affected helpers are also affected. It will be a chain reaction. Simultaneously, the helper who cannot achieve communication with the helper who will leave by multi-hop D2D communications will not be affected, such as helper 4.

In the other hand, when Helper 5 leaves the network, no helper will change the caching content placement. As such, another conclusion can be drawn. It is that only the departure of the helper who caches the more popular content will affect the caching content placement of the other involving helpers.

With the analysis above, we define a $F \times N$ matrix X that $x_{ji} \in \{0,1\}$. The problem of the caching content placement can be formulated as follows:

$$\begin{aligned} \max \quad & \sum\nolimits_{k=1}^{K} \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{F} r_{ki} x_{ji} P_j \\ s.t. \quad & \sum\nolimits_{j=1}^{F} x_{ji} \le M, \forall h_i \in \mathcal{H} \\ & \sum\nolimits_{i:h_i \in H(u_k)} x_{ji} \le 1, \forall u_k \in \mathcal{U}, \forall f_j \in \mathcal{F} \end{aligned} \quad \textbf{(6)}$$

In the above model, we try to solve the problem of caching content placement from the perspective of the whole network, and, the sum of the file download rate is set as the objective function. It is a monotone submodular function which is proved in the following section. Two kinds of constraints should be considered in our model. The first one is the constraint of limited storage capacity. We will prove that it is a matroid constraint in the following section. What's more, the second kind of constraints are set to take full use of the storage capacity of helpers which are multiple knapsack constraints.

3 The Monotone Submodular Set Function Over One Matroid and Multiple Knapsack Constraints

In this section, we analyze our proposed problem of the caching content placement in detail. Firstly, we define a ground set ε as

$$\varepsilon = \left\{ X \middle| x_{ji} \in \{0,1\}, \sum_{j=1}^{F} \sum_{i=1}^{N} x_{ji} = 1, j = 1, \cdots, F, i = 1 \cdots, N \right\} \quad \textbf{(7)}$$

For simplify, we present the ground set ε as $\varepsilon = \{\varepsilon_1^1, \varepsilon_2^1, \cdots, \varepsilon_F^1, \cdots, \varepsilon_1^N, \varepsilon_2^N, \cdots, \varepsilon_F^N\}$. ε_j^i denotes that $x_{ji} = 1$ and $\sum_{j=1}^{F} \sum_{i=1}^{N} x_{ji} = 1$, namely, the file f_j is cached in the helper h_i. ε can be divided into N disjoint subsets, $\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_N$, where $\varepsilon_i=\{\varepsilon_1^i,\varepsilon_2^i,\cdots,\varepsilon_F^i\}$ is the set of the indicators whether the files are cached in the helper h_i.

Secondly, some important definitions for the matroid and the submodular function are shown in the following according to the presentation in [19] and [16].

Definition 1. A matroid $\mathcal{M}$ is a tuple $\mathcal{M}=(\mathcal{S},\mathcal{G})$, where $\mathcal{S}$ is a finite ground set and $\mathcal{G}\subseteq 2^{\mathcal{S}}$ is a collection of independent sets, such that

(1) $\mathcal{G}$ is nonempty, in particular, $\varnothing\in\mathcal{G}$.

(2) $\mathcal{G}$ is downward closed; i.e., if $\mathcal{Y}\in\mathcal{G}$ and $\mathcal{X}\subseteq\mathcal{Y}$, then $\mathcal{X}\in\mathcal{G}$.

(3) If $\mathcal{X},\mathcal{Y}\in\mathcal{G}$, and $|\mathcal{X}|<|\mathcal{Y}|$, then $\exists y\in\mathcal{Y}\setminus\mathcal{X}$ such that $\mathcal{X}\cup\{y\}\in\mathcal{G}$.

Definition 2. Let $\mathcal{S}$ be a finite ground set. A set function $f:2^{\mathcal{S}}\to R$ is submodular function if for all sets $\mathcal{A},\mathcal{B}\in\mathcal{S}$,

$$f(\mathcal{A})+f(\mathcal{B})\ge f(\mathcal{A}\cup\mathcal{B})+f(\mathcal{A}\cap\mathcal{B}) \tag{8}$$

Definition 3. Let $f_{\mathcal{A}}(i)=f(\mathcal{A}\cup i)-f(\mathcal{A})$ denote the marginal value of an element $i\in\mathcal{S}$ with respect to a subset $\mathcal{A}\subset\mathcal{S}$. Then f is submodular function if for all $\mathcal{A}\subset\mathcal{B}\subset\mathcal{S}$ and for all $i\in\mathcal{S}\setminus\mathcal{B}$ we have

$$f_{\mathcal{A}}(i)\ge f_{\mathcal{B}}(i) \tag{9}$$

Followed, several theorems must be proved.

Theorem 1. The constraint $\sum_{j=1}^{F}x_{ji}\le M$ can be written as a partition matroid on the ground set ε.

Proof: In the constraint $\sum_{j=1}^{F}x_{ji}\le M$, a cache placement is expressed by the matrix X. The set $\mathcal{X}\subseteq\varepsilon$ is the cache placement set and $\varepsilon_j^i\in\mathcal{X}$ if and only if $x_{ji}=1$. It is obvious that the nonzero elements of the i th column of X correspond to the elements in $\mathcal{X}\cap\varepsilon_i$. As a result, the constraints on the cache capacity of the helpers $\sum_{j=1}^{F}x_{ji}\le M$ can be shown as $\mathcal{X}\subseteq\mathcal{G}$, where

$$\mathcal{G}=\{\mathcal{X}\subseteq\varepsilon:|\mathcal{X}\cap\varepsilon_i|\le M,\forall i=1,\cdots,N\} \tag{10}$$

$\mathcal{G}$ is all feasible solutions which meet the constraint on the storage capacity of the helpers $\sum_{j=1}^{F}x_{ji}\le M$. Comparing $\mathcal{G}$ and the example of the definition of the matroid in [16], we can see that the constraints on the cache capacity of the helpers $\sum_{j=1}^{F}x_{ji}\le M$ can form a partition matroid with $l=N$ and $k_i=M$, for $i=1,\cdots,N$. The partition matroid is shown as $\mathcal{M}=(\varepsilon,\mathcal{G})$.

Theorem 2. The objective function in Eq. (6) is a monotone submodular set function.

Proof: To simplify, we present the objective function as $f(\mathcal{X})=\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{F}r_{ki}x_{ji}P_j$. In the following, we prove Theorem 2 by two steps.

Firstly, it is obvious that the objective function in Eq. (6) is a monotone function. The proof is shown as follows.

We set $\mathcal{A}\subset\varepsilon$ and $\mathcal{A}'=\mathcal{A}\cup a\subset\varepsilon$ where $\{a\}$ is $\{\varepsilon_{j'}^{i'}\}$ which represents $x_{j'i'}=1$, and $\{\varepsilon_{j'}^{i'}\}$ does not belong to the set $\mathcal{A}$. Obviously, $\mathcal{A}\subseteq\mathcal{A}'$ Then, we can get $f(\mathcal{A})$ and $f(\mathcal{A}')$ as follows, respectively.

$$f(\mathcal{A})=\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{F}r_{ki}x_{ji}P_j \tag{11}$$

$$\begin{aligned}f(\mathcal{A}')&=f(\mathcal{A}\cup\{a\})\\&=\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{F}r_{ki}x_{ji}P_j+\sum_{k=1}^{K}r_{ki'}x_{j'i'}P_{j'}\end{aligned} \tag{12}$$

Due to $r_{ki'}\ge0$, $P_{j'}>0$ and $x_{j'i'}\ge0$, we can conclude that $f(\mathcal{A}')\ge f(\mathcal{A})$. As such, the objective function $f(\mathcal{X})=\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{F}r_{ki}x_{ji}P_j$ in Eq. (6) is a monotone function.

Secondly, we prove that it is also a submodular function. That is, for any set $\mathcal{V}\subset\mathcal{W}\subset\varepsilon$ and $e\in\varepsilon\setminus\mathcal{W}$, when $f_{\mathcal{V}}(e)\ge f_{\mathcal{W}}(e)$ is met, f is submodular.

To show the proof clear, we denote $f(\mathcal{X})$ as follows.

$$\begin{aligned}f(\mathcal{X})&=\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{F}r_{ki}x_{ji}P_j\\&=\sum_{k=1}^{K}\sum_{X}r_{ki}x_{ji}P_j\end{aligned} \tag{13}$$

And then, we can show $f_{\mathcal{V}}(e)$ and $f_{\mathcal{W}}(e)$ as follows.

$$\begin{aligned}f_{\mathcal{V}}(e)&=f(\mathcal{V}\cup e)-f(\mathcal{V})\\&=\sum_{k=1}^{K}\sum_{\mathcal{V}\cup e}r_{ki}x_{ji}P_j-\sum_{k=1}^{K}\sum_{\mathcal{V}}r_{ki}x_{ji}P_j\\&=\sum_{k=1}^{K}r_{ki'}x_{j'i'}P_{j'}\end{aligned} \tag{14}$$

$$\begin{aligned}f_{\mathcal{W}}(e)&=f(\mathcal{W}\cup e)-f(\mathcal{W})\\&=\sum_{k=1}^{K}\sum_{\mathcal{W}\cup e}r_{ki}x_{ji}P_j-\sum_{k=1}^{K}\sum_{\mathcal{W}}r_{ki}x_{ji}P_j\\&=\sum_{k=1}^{K}r_{ki'}x_{j'i'}P_{j'}\end{aligned} \tag{15}$$

Namely,

$$f_{\mathcal{V}}(e)=f_{\mathcal{W}}(e) \tag{16}$$

As a result, it is proved that the objective function in Eq. (6) is a monotone submodular set function.

Further, it is obvious that there are KF knapsack constraints $\sum_{i:h_i\in H(u_k)}x_{ji}\le1$. As a result, the model is the maximization of a monotone submodular set function over one matroid and multiple knapsack constraints.

Theorem 3. The problem of the caching content placement in Eq. (6) is NP hard.

Proof: We consider an instance of Eq. (6) where the number of files is $F=1$. Thus, we have $|\varepsilon|=N$ and P_j degenerates to P. Also, the original problem degenerates to

$$\begin{aligned} \max \quad & \sum_{k=1}^{K}\sum_{i=1}^{N} r_{ki} x_i P \\ s.t. \quad & \sum\nolimits_{i:h_i\in H(u_k)} x_i \le 1, \forall 1\le k\le K \end{aligned} \tag{17}$$

where r_{ki} is the capacity of the link between requester u_k and helper h_i. x_i is an indicator whether the file is cached in the helper h_i.

To prove that the original problem is NP-hard, we show that the problem specialized to the instance is NP-hard firstly. We prove that Eq. (17) is NP-hard via contradiction. Suppose that an efficient algorithm (with a complexity polynomial in N) exists that can optimally solve Eq. (17) for any input N and K. However, since H is different for different requesters for any input N and K, all possible cases should be computed to obtain the final solution. This would then contradict the assumption of the complexity polynomial in N. Namely, Eq. (17) is NP hard and further Eq. (6) is NP hard.

Obviously, it is impossible to obtain the optimal solution of Eq. (6). Luckily, we prove that the problem is the maximization of a monotone submodular function over one matroid and multiple knapsack constraints. About such kind of problem, there exists a very good characteristic [16]. It is that a suboptimal solution which yields a constant-factor approximation to the original problem can be obtained with polynomial complexity by utilizing the greedy algorithm. The constant-factor and the complexity will be given in the following Section.

4 The Caching Content Placement Algorithm Based on Greedy Algorithm

In the above section, we have proved that the problem of the caching content placement is the maximization of a monotone submodular function over one matroid and multiple knapsack constraints. According to the presentation in [18], the caching content placement algorithm based on greedy algorithm can be designed to obtain a suboptimal solution.

In this section, we define $\mathcal{I}=\left\{X\subseteq\varepsilon:\sum\nolimits_{j=1}^{F}x_{ji}\le M, \forall i=1,\cdots,N;\sum\nolimits_{i:h_i\in H(u_k)}x_{ji}\le 1,\forall k=1,\cdots,K,\forall j=1,\cdots,F\right\}$ as the domain of definition firstly. Then, we show the caching content placement algorithm based on greedy algorithm which is called CCPAGA for simplify as Algorithm 1. The work of computation in step 3 can be completed by all helpers. And a central node can be set to collect all results of helpers and determine the final strategy in each iteration by comparing all results.

Algorithm 1. The Caching Content Placement Algorithm Based on Greedy Algorithm (CCPAGA)

1. Initialization: Let $\mathcal{S}=\varnothing$.
2. **Repeat:**
3. Determine $\hat{e}=\underset{\substack{e\in\varepsilon\setminus\mathcal{S};\mathcal{S}\cup e\in\mathcal{I}\\ \sum_{j=1}^{F}x_{ji}\le M;\sum_{i:h_i\in H(u_k)}x_{ji}\le 1}}{\arg\max} f(X)$, and set $\mathcal{S}\leftarrow\mathcal{S}\cup\hat{e}$.
4. **Until** $\varepsilon\setminus\mathcal{S}=\varnothing$ or $\varepsilon\setminus\mathcal{S}\notin\mathcal{I}$.
5. Output $\mathcal{S}$.

In the following, we discuss the complexity of the proposed caching content placement algorithm based on greedy algorithm and the performance of the final solution on theory.

Theorem 4. The complexity of the proposed caching content placement algorithm based on greedy algorithm is $O\left(F^2N^2+FN\right)$.

Proof: According to the presentation of the caching content placement algorithm based on greedy algorithm, it is obvious that the final solution can be obtained in at most FN iterations. And in each iteration, we need to select the best solution from at most FN solutions by comparing the corresponding rates. It is obvious that the complexity of the proposed caching content placement algorithm based on greedy algorithm is $O\left(F^2N^2+FN\right)$.

Theorem 5. The proposed caching content placement algorithm based on greedy algorithm yields a constant-factor $\frac{1}{2+KF}$ approximation to the problem in this paper.

Proof: In the constraint $\sum\nolimits_{i:h_i\in H(u_k)}x_{ji}\le 1$, a cache placement of the helpers which are adjacent to the requester u_k is expressed by the matrix X_k. The set $\mathcal{X}_k\subseteq\varepsilon$ is the cache placement set of the helpers which are adjacent to the requester u_k and $\varepsilon_j^i\in\mathcal{X}_k$ if and only if $x_{ji}=1$. We set $\varepsilon^j=\{\varepsilon_j^1,\varepsilon_j^i,\cdots,\varepsilon_j^N\}$. It is obvious that the nonzero elements of the jth line of $\mathcal{X}_k$ correspond to the elements in $\mathcal{X}_k\cap\varepsilon^j$. As a result, the constraints $\sum\nolimits_{i:h_i\in H(u_k)}x_{ji}\le 1$ can be shown as $\mathcal{X}_k\subseteq\mathcal{G}_k$, where

$$\mathcal{G}_k=\left\{\mathcal{X}_k\subseteq\varepsilon:\left|\mathcal{X}_k\cap\varepsilon^j\right|\le 1,\forall j=1,\cdots,F\right\} \tag{18}$$

It is obvious that the knapsack constraint $\sum\nolimits_{i:h_i\in H(u_k)}x_{ji}\le 1$ is a matroid constraint indeed. As a result, the proposed caching content placement algorithm based on greedy algorithm yields a constant-factor $\frac{1}{2+KF}$ approximation to the problem in this paper [18].

5 Simulation and Discussion

To verify the performance of our proposed caching content placement algorithm based on greedy algorithm, some simulations are done. Some parameters involved in the proposed algorithm are set in Table 1. All simulations are done based on the parameters in Table 1 without special explanation. Besides, in order to validate the performance of the proposed caching content placement based on greed algorithm, another caching content placement scheme is designed. This caching content placement scheme is designed to let all helpers cache the most popular files and the storage space of every helper is run out. Such an algorithm was mentioned in [9] to be an optimal caching content placement scheme when every wireless device has only access to an exactly one helper. For simplify, we call it First First algorithm. The detail is shown in Algorithm 2. In addition, we also consider the Random Caching in which each content is cached by any helper with the equal probability M/F.

Table 1. System parameters

The number of requesters	$K=10$
The number of helpers	$N=15$
The number of files	$F=25$
The cache capacity constraint	$M=5$
The total transmit power of each node	$p=0.3W$
The cell radius	$R=100m$
The noise power	$\sigma^2=-80dBm$
The requesters' location	Uniformly distributed in $[0,R]$
The helpers' location	Uniformly distributed in $[0,R]$
The path loss exponent	$\alpha=3$
The demand dominance factor of the Zipf distribution	$s=0.56$

Algorithm 2. The First First Algorithm

1. Initialization: Let $\mathcal{S}=\varnothing$.
2. **Repeat:**
3. For any $e\in\varepsilon$,
if $j\le M$,
set $x_{ji}=1$ and $\mathcal{S}\leftarrow\mathcal{S}\cup e$.
4. **Until** $|\mathcal{S}|=MN$. ($|\mathcal{S}|$ denotes the number of matrices in $\mathcal{S}$.)
5. Output $\mathcal{S}$.

In Figure 4, Figure 5 and Figure 6, we compare the proposed caching content placement algorithm based on greedy algorithm with the First First algorithm and Random Caching on the cache hit rate, the number of the available files for each requester and the file download rate. It is shown that with CCPAGA, the requesters can enjoy the higher cache hit rate, more files and larger file download rate. In such a case that the total number of the files F is much larger than the cache capacity M, it is obvious to obtain a higher cache hit rate with the constraint $\sum_{i:h_i\in H(u_k)}x_{ji}\le 1$ in our model. To summarize according to the Figure 4, Figure 5 and Figure 6, the model in this work is valuable and CCPAGA can do well.

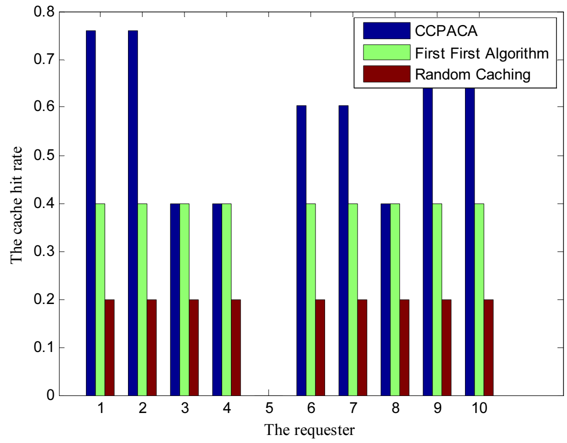


Figure 4. The cache hit rate of each requester for different algorithms when $F=25$ and $M=5$

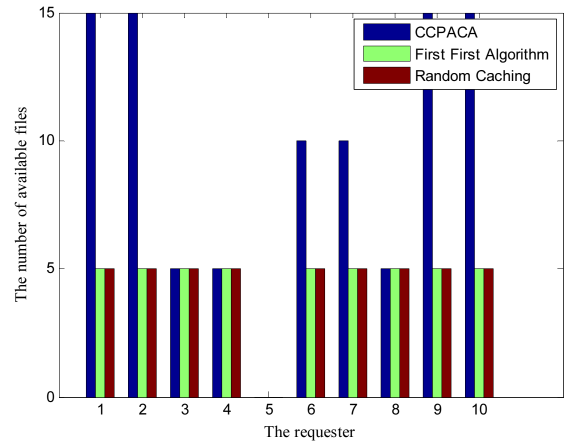


Figure 5. The number of the available files for each requester for different algorithms when $F=25$ and $M=5$

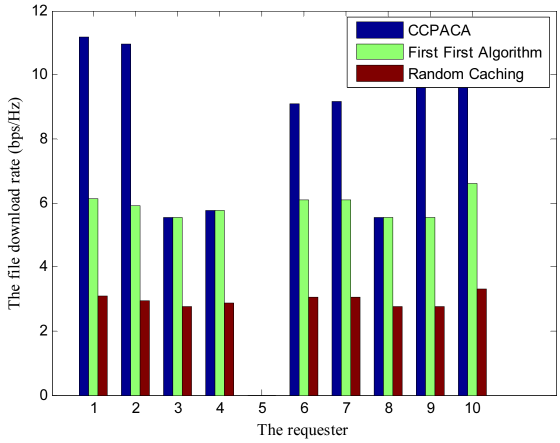


Figure 6. The file download rate of each requester for different algorithms when $F = 25$ and $M = 5$

In Figure 7 and Figure 8, we consider a special case that the cache capacity is large enough to meet all the files. In other words, just several files are interesting for the requesters. Specifically, in the two simulations, we set $F = M = 20$ to present the special case. In Figure 7, we can see that the file download rates of the requester 1, the requester 2 and the requester 6 with CCPAGA are lower than these with the First First algorithm and Random Caching. However, we also can see that just four helpers are used to provide the service of file download with CCPAGA in Figure 8. Obviously, the phenomenon that just four helpers are used to cache the files appears because of the constraint $\sum_{i:h_i \in H(u_k)} x_{ji} \le 1$. As a result, to analyze the reason for the phenomenon in Figure 7, we do a simulation to obtain the Figure 9.

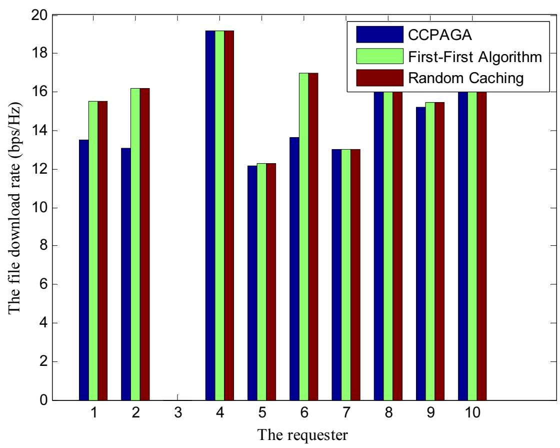


Figure 7. The file download rate of each requester for different algorithms when $F = M = 20$

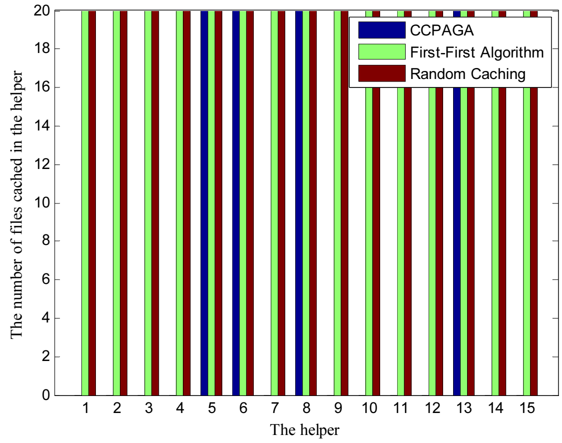


Figure 8. The file cached in each helper for different algorithms when $F = M = 20$

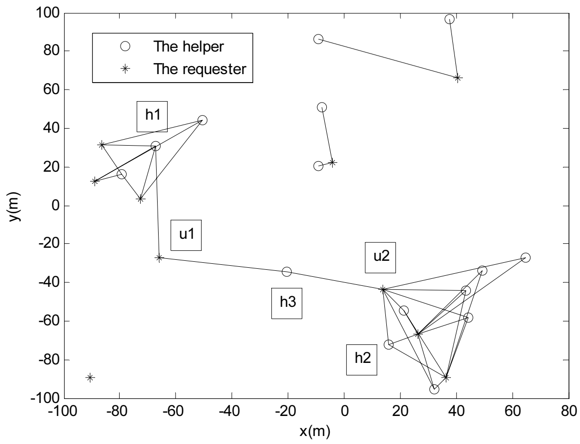


Figure 9. The locations of requesters and helpers

In Figure 9, the locations of all nodes are presented. Something should be noted. For example, h3 serves u1 and u2 at the same time. In addition, u1 is also served by h1 and u2 is also served by h2. The other nodes are ignored. According to CCPAGA and the second constraint of our model, the file will be cached in h3 only when $r_{u1,h3} + r_{u2,h3} > \max\left(r_{u1,h1}, r_{u2,h2}\right)$. And the result will not be changed even if $r_{u1,h1} > r_{u1,h3}$ and $r_{u2,h2} > r_{u2,h3}$ since it will create more benefits with fewer helpers to cache the files in h3. To sum up, there is no wrong with the final caching content placement obtained through CCPAGA. Although the file download rates of the requester 1, the requester 2 and the requester 6 with CCPAGA are lower than these with the First First algorithm, when we consider the result from another perspective, it is noticed that the efficient of the helper with CCPAGA is a valuable topic to discuss.

To discuss the efficiency of the helper with CCPAGA, we define the sum file download rate one helper can provide as the efficiency of the helper.

$$E = \frac{\sum_{k=1}^{K} FDR_k}{N_{helper}} \tag{19}$$

where N_{helper} denotes the number of helpers in which the files are cached.

In Figure 10, we discuss the efficiency of the helper with the Monte Carlo method in different cases in which the numbers of the requesters are 4, 7, 10, 13, 16. The simulation result shows that with the number of the requesters increasing, the efficiency of the helpers is higher and higher. In addition, it is obvious that the efficiency of the helper with CCPAGA is higher than that with the First First algorithm and Random Caching in all cases. The helpers will play better roles with CCPAGA comparing to the fixed caching content placement scheme of the First First algorithm and Random Caching.

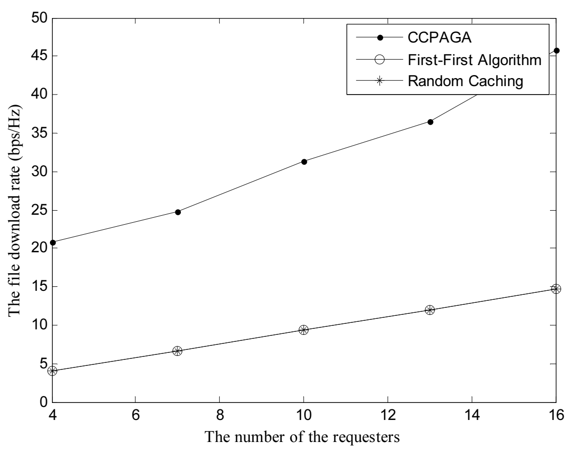


Figure 10. The file download rate one helper can provide vs. the number of requesters for different algorithms when $F = M = 20$ with the Monte Carlo method

The case $F \approx M$ is possible in fact. For example, in a limited space, some requesters with the same interests just are interested in several files. But luckily, the special case above is very few in fact. More cases in the fact we will face are $F \gg M$. In the following, we will consider different cases in which the number of requesters, the number of helpers and the demand dominance factor are different.

Firstly, we will discuss the affection of the demand dominance factor on the performance of CCPAGA, the First First algorithm and Random Caching in Figure 11. The performance of Random Caching is worst and is not affected by the demand dominance factor. The performance of CCPAGA is better than that of the First First algorithm when the demand dominance factor is small. In addition, the efficiency of the helper is higher with the increasing of the demand dominance factor under the two algorithms. But the advantage of CCPAGA on the efficiency of the helper is decreasing with the increasing of the demand dominance factor. Such a phenomenon is reasonable. No matter which algorithm is selected, the files which are the most popular will be cached. With the increasing of the demand dominance factor, the probability requesting for such files will be higher. There is no doubt that the efficiency of the helper is higher with the increasing of the demand dominance factor under the two algorithms. But similarly, with the increasing of the demand dominance factor, fewer and fewer files accounts for more and more requests. As a result, it is more and more meaningless to cache the files with smaller requesting probability. Specially, it will worsen CCPAGA more since more files with lower requesting probability will be cached with CCPAGA. However, in my opinion, as the above exposition on Zipf distribution shows, it is unnecessary to discuss the problem of the caching content placement with a too large demand dominance factor. Except our work, many other works also select a small value for the demand dominance factor, such as, 1 in [14], 0.56 in [16], 0.8 in [20], 0.5 and 2 in [21], and 0.6 in [22].

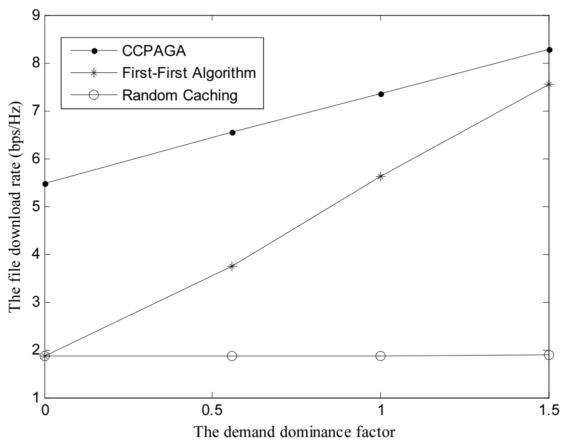


Figure 11. The file download rate one helper can provide vs. the demand dominance factor s for different algorithms when $F = 25$ and $M = 5$ with the Monte Carlo method

Secondly, we discuss the affection of the number of the helpers and the number of the requesters on the performance of the three algorithms. In Figure 12, some conclusions can be made. Firstly, with the number of the requesters increasing, the efficiency of the helpers will be enhanced. It is not hard to explain that with the number of the requesters increasing, a helper can provide service for more requesters and play a larger role. Secondly, more helpers will not enhance the efficiency of the helpers. For CCPAGA, more helpers mean larger storage space of the whole network. Followed, more files can be cached and provided for the requesters. However, it will enhance the sum file download rate less due to the lower requesting probability and result in the deterioration of the efficiency of the helpers. Similarly, for the First First algorithm and Random Caching, more helpers mean more helpers in bad locations will play no role. As a result, the efficiency of the helpers will be lower with the increasing of the number of the helpers. Thirdly, the performance of CCPAGA is better than that of the First First algorithm and Random Caching in all cases.

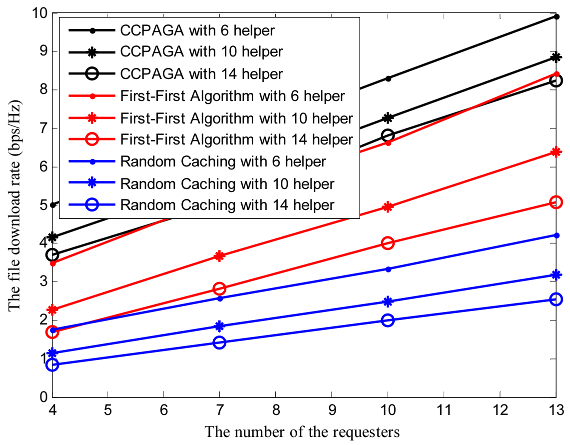


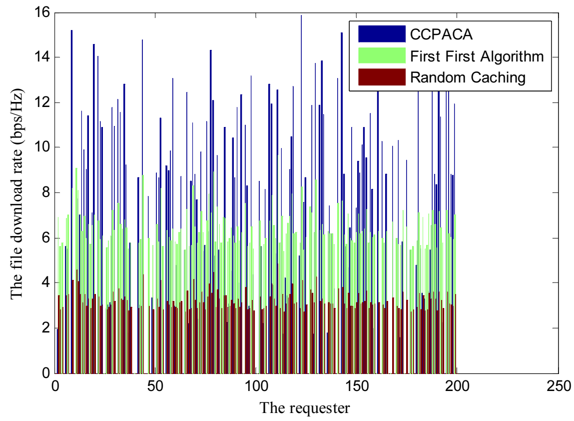
Figure 12. The file download rate one helper can provide vs the number of the requesters for different number of helpers and different algorithms when $F=25$ and $M=5$ with the Monte Carlo method

Further, we enlarge the simulation scale to well verify the validity of the proposed algorithm. In detail, we compare the caching content placement in three different scenarios in Figure 13.

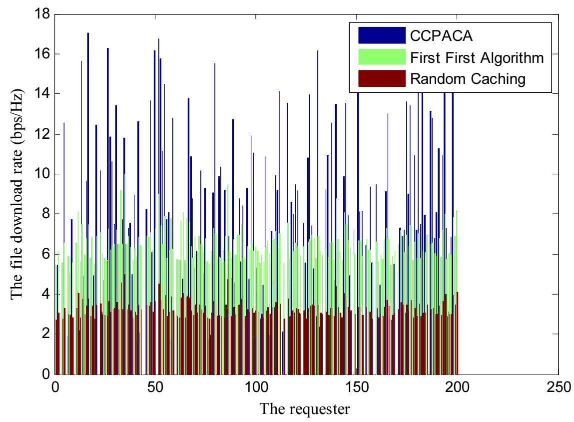
In Figure 13, we consider three different scenarios where $N=30$, $N=40$, $N=50$. We can see that the performance of our proposed CCPACA is better than that of First First Algorithm and Random Caching in all three algorithms. In other words, the simulation scale will affect the result of caching content placement, but the conclusion that the performance of our proposed CCPACA is better than that of First First Algorithm and Random Caching will not be affected.

6 Conclusion

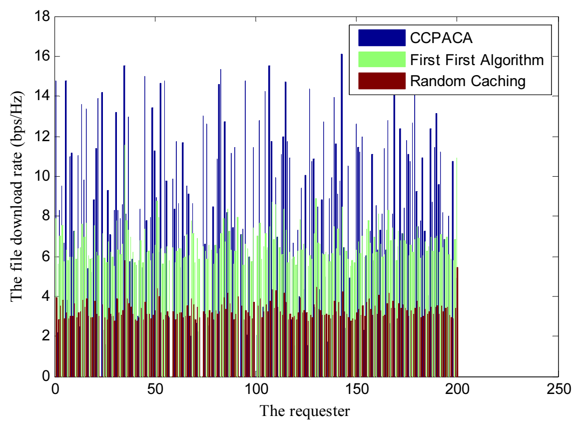
In this paper, we want to provide a better content download service for requesters and reduce the pressure of the base station. To achieve such a goal, we pay attention on the content caching based on D2D communications and the introduced issue of the caching content placement due to the proposal of content caching. In particular, we mean to maximize the sum file download rate by design a proper caching content placement. The problem is expressed as a monotone submodular function over one matroid and multiple knapsack constraints. Besides, the problem of the caching content placement is proved to be NP hard. A caching content placement algorithm based on greedy algorithm is proposed to obtain a suboptimal solution. It is proved that the complexity of the proposed caching content placement algorithm based



(a) $K=200$, $N=30$, $F=25$, $M=5$



(b) $K=200$, $N=40$, $F=25$, $M=5$



(c) $K=200$, $N=50$, $F=25$, $M=5$

Figure 13. Caching content placement in three different scenarios

on greedy algorithm is polynomial and it yields a constant-factor $\frac{1}{2+KF}$ approximation to the problem in this paper. The simulation results show that the efficiency of the helper is higher through our proposed caching content placement algorithm based on greedy algorithm.

Acknowledgments

This work is supported by the National Natural Science Foundation of China No. 61671474 and the Jiangsu Provincial Natural Science Foundation for Excellent Young Scholars No. BK20170089.

References

[1] C. Fang, F. R. Yu, T. Huang, J. Liu, Y. Liu, A Survey of Energy-Efficient Caching in Information-Centric Networking, *IEEE Communications Magazine*, Vol. 52, No. 11, pp. 122-129, November, 2014.

[2] D. Wu, L. Zhou, Y. Cai, Social-Aware Rate Based Content Sharing Mode Selection for D2D Content Sharing Scenarios, *IEEE Transactions on Multimedia*, Vol. 19, No. 11, pp. 2571-2582, November, 2017.

[3] L. Zhou, D. Wu, J. Chen, Z. Dong, Greening the Smart Cities: Energy-Efficient Massive Content Delivery via D2D Communications, *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 4, pp. 1626-1634, April, 2018.

[4] D. Wu, J. Wang, R. Q. Hu, Y. Cai, Energy-Efficient Resource Sharing for Mobile Device-To-Device Multimedia Communications, *IEEE Transactions on Vehicular Technology*, Vol. 63, No. 5, pp. 2093-2103, June, 2014.

[5] L. Zhou, D. Wu, Z. Dong, X. Li, When Collaboration Hugs Intelligence: Content Delivery over Ultra-Dense Networks, *IEEE Communications Magazine*, Vol. 55, No. 12, pp. 91-95, December, 2017.

[6] D. Wu, Y. Cai, R. Q. Hu, Y. Qian, Dynamic Distributed Resource Sharing for Mobile D2D Communications, *IEEE Transactions on Wireless Communications*, Vol. 14, No. 10, pp. 5417-5429, October, 2015.

[7] R. Wang, X. Peng, J. Zhang, K. B. Letaief, Mobility-Aware Caching for Content-Centric Wireless Networks: Modeling and Methodology, *IEEE Communications Magazine*, Vol. 54, No. 8, pp. 77-83, August, 2016.

[8] M. Sheng, C. Xu, J. Liu, J. Song, X. Ma, J. Li, Enhancement for Content Delivery with Proximity Communications in Caching Enabled Wireless Networks: Architecture and Challenges, *IEEE Communications Magazine*, Vol. 54, No. 8, pp. 70-76, August, 2016.

[9] N. Golrezaei, A. F. Molisch, A. G. Dimakis, 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, April, 2013.

[10] C. Xu, P. Zhang, S. Jia, M. Wang, G. M. Muntean, Video Streaming in Content-Centric Mobile Networks: Challenges and Solutions, *IEEE Wireless Communications*, Vol. 24, No. 5, pp. 157-165, October, 2017.

[11] J. Jiang, S. Zhang, B. Li, B. Li, Maximized Cellular Traffic Offloading via Device-to-Device Content Sharing, *IEEE Journal on Selected Areas in Communications*, Vol. 34, No. 1, pp. 82-91, January, 2016.

[12] D. Malak, M. Shalash, J. Andrews, Spatially Correlated Content Caching for Device-to-Device Communications, *IEEE Transactions on Wireless Communications*, Vol. 17, No. 1, pp. 56-70, January, 2018.

[13] M. Gregori, J. Gomez-vilardebo, J. Matamoros, D. Gunduz, Wireless Content Caching for Small Cell and D2D Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 34, No. 5, pp. 1222-1234, May, 2016.

[14] B. Chen, C. Yang, G. Wang, Cooperative Device-to-Device Communications with Caching, *IEEE 83th Vehicular Technology Conference*, Nanjing, China, 2016, pp. 1-5.

[15] X. Huang, G. Zhao, Z. Chen, Segment-Based Random Caching in Device-to-Device (D2D) Caching Networks, *International Symposium on Wireless Communication Systems*, Brussels, Belgium, 2015, pp. 731-735.

[16] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, G. Caire, FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers, *IEEE Transactions on Information Theory*, Vol. 59, No. 12, pp. 8402-8413, December, 2013.

[17] J. Qu, Y. Cai, J. Zheng, W. Yang, W. Yang, Y. Hu, Interference Mitigation Based on Intelligent Location Selection in a Canonical Communication Network, *Frequenz*, Vol. 70, No. 1-2, pp. 57-61, January, 2016.

[18] N. Prasad, H. Zhang, H. Zhu, S. Rangarajan, Multi-User MIMO Scheduling in the Fourth Generation Cellular Uplink, *IEEE Transactions on Wireless Communications*, Vol. 12, No. 9, pp. 4272-4285, September, 2013.

[19] G. Nemhauser, L. Wolsey, *Integer and Combinatorial Optimization*, Wiley, 1988.

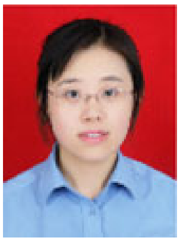
[20] C. Yang, Y. Yao, Z. Chen, B. Xia, Analysis on Cache-Enabled Wireless Heterogeneous Networks, *IEEE Transactions on Wireless Communications*, Vol. 15, No. 1, pp. 131-145, January, 2016.

[21] D. Malak, M. Al-Shalash, Device-to-Device Content Distribution: Optimal Caching Strategies and Performance Bounds, *IEEE International Conference on Communication Workshop*, London, UK, 2015, pp. 664-669.

[22] L. Zhang, M. Xiao, G. Wu, S. Li, Efficient Scheduling and Power Allocation for D2D-assisted Wireless Caching Networks, *IEEE Transactions on Communications*, Vol. 64, No. 6, pp. 2438-2452, June, 2016.

Biographies

Junyue Qu, received the B.S. degree and M.S. degree both from the College Communications Engineering, PLAUST, Nanjing, China in 2012 and 2015, respectively. He is pursuing for the Ph.D. degree in Army Engineering University of PLA, Nanjing, China. His current research interest is content sharing based on D2D communications.



Dan Wu received the B.S., M.S., and Ph.D. degrees from PLAUST, Nanjing, China, in 2006, 2009, and 2012, respectively. She is currently a Postdoctoral Researcher in Army Engineering University of PLA. Her research interests include resource allocation and management, game theory, cooperative communications, and wireless sensor networks.

Yanshan Long received the B.S. and M.S. degrees from PLAUST, Nanjing, China, in 2011 and 2014, respectively. She is currently pursuing PhD. degree in the Institute of Communications Engineering, PLA University of Science and Technology. Her research interests include D2D communications, wireless caching network and stochastic geometry theory.

Wendong Yang received the B.S. degree and the Ph.D. degree both from College of Communications Engineering, PLAUST, Nanjing, China in 2004 and 2009, respectively. Since 2009, he has been with Army Engineering University of PLA. His current research interest includes MIMO systems, OFDM systems, cooperative communications and cognitive radio.

Yueming Cai received the B.S. degree in physics from Xiamen University, Xiamen, China, in 1982, the M.S. degree in microelectronics engineering and the Ph.D. degree in communications and information systems from Southeast University, Nanjing, China, in 1988 and 1996, respectively. His research interests include MIMO, OFDM, cooperative communications, and WSN.