Delay-Oriented Caching Strategies in D2D Mobile Networks

Ruijin Sun, *Student Member, IEEE*, Ying Wang, *Member, IEEE*, Ling Lyu, *Student Member, IEEE*, Nan Cheng, *Member, IEEE*, Shan Zhang, *Member, IEEE*, Tingting Yang, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Caching-enabled device-to-device (D2D) networks have the potential to make mobile users directly fetch requested files from nearby users, resulting in low network delay. In addition, user mobility can increase the communication chances among different users, and therefore, the network delay can be further effectively reduced by proper designing the caching strategy. In this paper, mobility-aware caching strategies in D2D networks are studied to minimize the network delay. Specifically, based on the inter-contact user mobility model, the expression of the average file delivery delay is analytically obtained. Considering the limited cache capacity, a delay minimization cache placement problem considering the user mobility is investigated. To optimally solve this nonlinear integer programming problem, we reformulate it as a multistage decision problem. According to the recursive relationship between adjacent stages, dynamic programming is adopted to obtain the optimal mobility-aware caching strategy stage-by-stage. Furthermore, to lower the complexity, we also demonstrate that the original problem can be recasted as a monotone submodular function maximization problem over a matroid constraint. Then, a low-complexity greedy mobility-aware caching strategy with $(1-1/e)$-optimality performance guarantee is put forward. Numerical results show that, in the scenario with high user mobility, the file delivery delay can be reduced by 47% with our proposed mobility-aware caching strategy, as compared with the most popular caching. Furthermore, the superiority of the proposed caching strategy is verified by real-world data set.**

***Index Terms*—Cache placement, file delivery delay, inter-contact model, nonlinear integer problem.**

Manuscript received March 23, 2020; accepted May 17, 2020. Date of publication May 21, 2020; date of current version August 13, 2020. This work was supported in part by the National Nature Science Foundation of China under Project 61431003, and in part by the China Scholarships Council. The review of this article was coordinated by Dr. K. Bian. *(Corresponding author: Ying Wang.)*

Ruijin Sun and Ying Wang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: sunruijin@bupt.edu.cn; wangying@bupt.edu.cn).

Ling Lyu is with the School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China (e-mail: linglyu@yeah.net).

Nan Cheng is with the School of Telecommunication Enginnering, Xidian University, Xi'an 710071, China (e-mail: wmchengnan@gmail.com).

Shan Zhang is with the School of Computer Science and Technology, Beihang University, Beijing 100191, China (e-mail: zhangshan18@buaa.edu.cn).

Tingting Yang is with the School of Electrical Engineering and Intelligentization, Dongguan University of Technology, Dongguan 523000, China, and also with the Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: yangtingting820523@163.com).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TVT.2020.2996238

I. Introduction

THE last decade has witnessed the explosive increase of the wireless traffic data, which is brought by the widespread proliferation of smart mobile phones and the rapid development of mobile Internet. It has been reported that the mobile data traffic will reach 77.5 exabytes per month by 2022, being 7 times than that of 2017 [1]. Furthermore, newly emerging services, such as online games, high-definition video streaming, augmented realities and so on, also put emphasis on the low network delay. To enhance the network capacity, ultra-dense networks with spectrum reuse are proposed [2], [3]. However, this network architecture poses a huge traffic pressure on the capacity-limited backhaul links and does not provide significant improvement on the network delay.

By pre-caching popular contents at BSs or mobile users in advance, wireless edge caching can avoid the redundant transmission of the same content on the backhaul links and thus effectively relieve the backhaul pressure [4]. Besides, with such edge caching, users can access their required contents directly from much closer BSs or adjacent mobile users, which can remarkably decrease the content retrieval delay and greatly improve the user experience. According to the location of the content caching, wireless edge caching can be divided into caching at BSs and caching at mobile users [5].

Compared with the BS caching [6], caching at mobile users with the aid of device-to-device (D2D) communications [7], [8] can further relieve the rate pressure of the wireless interface by offloading the network traffic to D2D links. Besides, as bringing contents closer to users, caching at mobile users also leads to lower network delay [9], [10]. Thus, lots of works on D2D caching have been carried out. From the time-scale viewpoint, based on content request delay information, a joint BS pushing and mobile user caching policy is proposed to find when and which content to be pushed by a BS and cached by a user [11]. This study is extended from a single user scenario to a more general multi-user scenario [12]. Then, a efficient cache placement algorithm of mobile users is proposed in [13] to minimize the average transmission delay. From the perspective of spatial stochastic networks, in [14], with the Gauss-Poisson process, the analytical expression of the content retrieval delay in D2D networks is derived. In [15], a probabilistic caching strategy for mobile users is investigated to maximize the density of successful receptions with the consideration of the reliability of D2D links. In [16], users with different file preferences

are divided into different interest groups and the file caching probabilities for different groups are optimized to maximize the probability of successful file discovery.

However, the user mobility, an intrinsic characteristic of mobiles users, is neglected in these works. With the user mobility, the locations of users dynamically change over time and users may have more chances to connect with different BSs and users, which can increase the probability of acquiring requested files. To exploit this benefit, the mobility-aware cache placement design, which is still in its infant stage, has started to draw lots of attentions [17]–[26]. By modelling the user movements that moving users access to different small-cell BSs at different times as a Markov chain, Poularakis *et al.* designed a cache placement at small-cell BSs to decrease the traffic load of the macro-cell [19]. Based on the Markov decision process, Qiao *et al.* investigated the caching strategy at on-road BSs to improve the user received video quality in high-mobility highway communication scenarios [20]. By modelling the dwelling time of a vehicle on a BS as Erlang distribution, a mobility-aware proactive content pushing is also investigated in the vehicular communication networks [21]. Nevertheless, these user mobility models for the BS caching is not suitable for D2D caching networks, where both the receiving users and the transmitting users are moving.

To characterize the user mobility in D2D networks, an inter-contact model is proposed [22]. Specifically, with the user mobility, the event that user pairs move into or move out each other's communication range refers as the contact or the inter-contact. The contact duration and the inter-contact duration respectively follow corresponding stochastic distributions modeled by the realistic human mobility. Based on this user mobility model, mobility-aware caching strategies for mobile users are put forward to increase the data offloading ratio [23], [24]. In [25], the mobility-aware caching strategy and transmission power are jointly optimized to improve the cache hit ratio. Supposing that the cost of users accessing required files from nearby users is much lower than that from BSs, cost-optimal mobility-aware caching strategies are designed [26].

In addition to the cache hit ratio and the network cost, newly emerging video-related services also urgently require the low network delay. This trend makes the network delay become a more significant metric in the cache placement design. Although the delay-aware caching of mobile users has been investigated in [11]–[14], all these works fail to take the user mobility into consideration. As the user mobility can increase the communication chances between different users, caching files at users with high mobility may facilitate the D2D file sharing and reduce the network delay. Therefore, it is necessary to figure out how to design the delay-optimal mobility-aware cache placement in D2D networks and how much the user mobility can help to reduce the network delay in D2D caching.

Motivated by this, this paper aims to study the delay-oriented cache placement problem in D2D networks considering the user mobility. To characterize the user mobility in D2D networks, the popular inter-contact model [22] is adopted, where the inter-contact duration follows the exponential distribution. Then, the mobility-aware average file delivery delay in D2D caching networks is analytically derived. Consequently, with the limited cache capacity, a delay-oriented mobility-aware cache placement problem is originally investigated. To handle this nonlinear integer programming (NLIP) problem, combinatorial optimization methods are adopted. Firstly, by equivalently reformulating the original cache placement problem as a multi-stage decision problem and finding the recursive relationship between adjacent stages, dynamic programming (DP) is adopted to achieve the optimal caching strategy by solving the problem stage-by-stage. Then, to lower the complexity, the delay-related objective function in the cache placement problem is demonstrated to have the monotone submodular property. To prove this, mathematical techniques exploiting the structure of the delay function are proposed, as the feature of the derived mobility-aware delay function in this paper is quite different from that of other performance metrics in the literature [23], [24]. With this property, the original problem is equivalently reformulated as a monotone submodular function maximization problem over matroid constraint. Thus, a greedy caching strategy is put forward, which can achieve $(1 - 1/e)$ optimality with linear complexity. Simulation results show that, with the almost same delay performance, the runtime of the proposed greedy caching strategy is remarkably shorter than that of the DP-based optimal caching strategy. This convincingly verifies the effectiveness of the proposed greedy caching strategy. Furthermore, with the high user mobility, the average file delivery delay can be decreased by 47% with the proposed mobility-aware caching, as compared with the most popular caching strategy. More importantly, the effectiveness of the proposed greedy caching strategy is verified by the real data set. The main contributions of this paper are summarized as follows.

- To figure out whether exploiting the user mobility information in D2D caching design can help to reduce the network delay, the delay-oriented cache placement problem considering the user mobility in D2D caching networks is originally studied in this paper.
- By characterizing the user mobility in D2D networks as the inter-contact model, the average file delivery delay is analytically derived. Then, considering the limited cache capacity, a delay minimization mobility-aware cache placement problem is formulated.
- The optimal caching strategy is achieved by adopting DP to deal with the formulated NLIP delay-oriented mobility-aware cache placement problem. Specifically, the original problem is first reformulated as a multi-stage decision problem and then optimally solved stage-by-stage according to the recursive relationship between adjacent stages.
- A greedy caching strategy with linear complexity is proposed, which has $(1 - 1/e)$-optimality performance guarantee. This is achieved by skillfully proving that the original problem can be equivalently reformulated as a monotone submodular function maximization problem over matroid constraint.
- Both simulation results and real-world data set show that our proposed mobility-aware caching strategies have the advantage over other caching strategies in terms of the average file delivery delay, especially when the user moving speed is high.

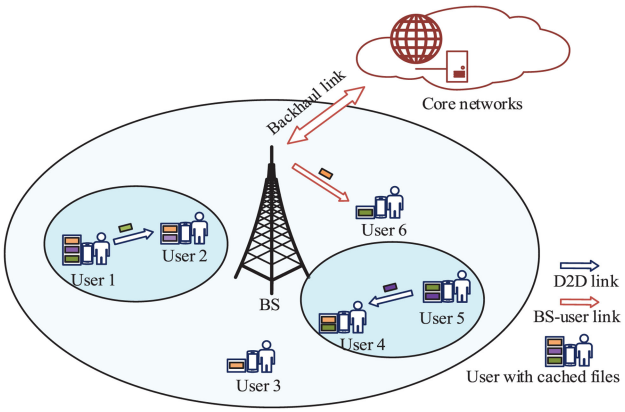


Fig. 1. The system model where moving users are sharing cached files via D2D links.

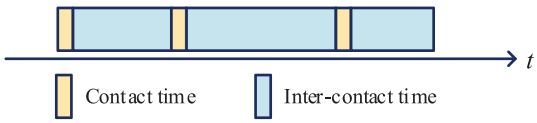


Fig. 2. The user contact process.

The remainder of the paper is organized as follows. Section II gives the system model and problem formulation. Section III and IV respectively propose the optimal caching strategy and the sub-optimal greedy caching strategy with low complexity. In Section V, simulation results are shown. Finally, Section VI makes a conclusion.

Notation: Bold lower case letters and bold upper case letters are used to denote column vectors and matrices, respectively. $\mathbb{E}\{x\}$ is the expectation of x and $\lceil x \rceil$ is the ceiling function with $x = \min\{n \in \mathbb{z} | n \geq x\}$.

II. System Model and Problem Formulation

A. System Scenario and User Mobility Model

Consider a 5G network where U mobile users are within the coverage of one BS, as depicted in Fig. 1. Let $\mathcal{U} = \{1, 2, ...U\}$ represent the user index set. Each user i, $i \in \mathcal{U}$, has a limited storage capacity C_i to cache popular files. Mobile users move randomly and can acquire their required files from others via D2D communications when they move into each other's communication range.

To characterize the user mobility pattern, the widely used inter-contact model [22] is also adopted in this paper. In particular, as shown in Fig. 2, the timeline of a user contact process alternatively consists of the contact time and the inter-contact time. The former is the time that two users are in the communication range and may share required files via D2D links. The latter is the time between two consecutive contacts. By analyzing with the real-world user traces [22], the inter-contact time between users i and j follows exponential distribution with parameter μ_{ij}. μ_{ij} is the contact rate denoting the average number of contacts between user i and j per unit time, which is shown to be linearly related to the user moving speed [23]. Accordingly, as the same with the recent literature [23], [26], the contact process between any two users i and j in this paper is modeled as a Poisson process with intensity μ_{ij}. Due to the randomness of the user mobility, it is reasonable to assume that contact processes of different user pairs are independent.

B. Cache Placement and Delivery Model

Consider a content library with F files and the set of these files is denoted as $\mathcal{F} = \{1, 2, ..., F\}$, each with size S. For the cache placement, the entire file is completely cached or not by mobile users. Let $\boldsymbol{X} = [x_{if}]^{U \times F}$ be the cache placement matrix, where $x_{if} \in \{0, 1\}$ denotes the discrete cache placement indicator for file f in user i. For efficient transmission, each file is further split into N segments with size $s = S/N$. During each user contact, $N_c (N_c \leq N)$ segments can be successfully delivered owing to the limited transmission rate.

In the scenario, each user requests desired file randomly and independently based on its file popularity p_{if}, which follows the famous Zipf distribution [27]. When user i requests file f, it will first find the required file segments from its own cache. If file f is not found, user i will require file f from the encountered users via D2D links. After a deadline T_D, if the whole file f is not successfully received by user i via D2D links, the undelivered file segments will be directly transmitted by the BS. Note that the deadline is to ensure that all users can finally access to their required files.

C. Average File Delivery Delay

The average file delivery delay is defined as the average waiting time of retrieving one file from the network. Since only several number of file segments can be delivered via each user contact, to obtain the average file delivery delay, we need to first derive the probability of the number of successfully delivered file segments via D2D communications.

Let the timeline start from user i submitting its requested file f. Note that the user contact process of users i and j follows Poisson with intensity μ_{ij}. The inter-contact time of user i first encountering user j, w^I_{ijf}, follows exponential distribution with parameter μ_{ij}. From the network perspective, the inter-contact time of user i meeting the first user in D2D networks is given as $t^{I_1}_{if} = \min_{j \in \mathcal{U}, x_{jf}=1} w^I_{ijf}$. As the contact process of different user pairs are independent, the first inter-contact time $t^{I_1}_{if}$ is an exponential distribution with parameter $\lambda_{if} = \sum_{j \in \mathcal{U}, x_{jf}=1} \mu_{ij} = \sum_{j \in \mathcal{U}} x_{jf} \mu_{ij}$. The contact rate μ_{ii} can be treated as infinity, which indicates that users i can meet itself with probability 1. In other words, user i can immediately acquire the required file from the its own storage if the file is cached by itself. Similarly, the l-th $(l > 1)$ inter-contact time, $t^{I_l}_{if}$, also follows the same distribution. Hence, for the case user i requesting file f, the contact process between user i and all users in D2D networks is Poisson process with intensity λ_{if}.

Based on the Poisson contact process analyzed above, we can derive the probability of the number of successfully delivered file segments via D2D links. Firstly, we consider the case that user i can only fetch $k (0 \leq k < N)$ file segments of file f via

D2D communications within the deadline T_D. Since N_c file segments can be delivered within each contact, the successful transmission of k ($k \mod N_c = 0$) file segments means that there are $l_k = k/N_c$ contacts within the deadline T_D. According to the Poisson process, the number of contacts within T_D follows Poisson distribution with parameter $\lambda_{if} T_D$. Thus, the probability of k ($0 \le k < N$) file segments being delivered via D2D communications is expressed as

$$P(K_{if}=k)=\begin{cases}\frac{e^{-\lambda_{if}T_D}(\lambda_{if}T_D)^{l_k}}{l_k!}, & k \mod N_c = 0,\\ 0, & k \mod N_c \neq 0,\end{cases} \quad (1)$$

where K_{if} is a random variable denoting the number of file segments delivered within T_D. $P(K_{if}=k)=0(k \mod N_c \neq 0)$ is due to the fact that the case of k ($k \mod N_c \neq 0$) file segments being delivered via D2D communications will not occur.

Then, for the event that the whole file is successfully transmitted within T_D (i.e., $K_{if}=N$), at least $l_N = \lceil \frac{N}{N_c} \rceil$ user contacts are required, where $\lceil x \rceil$ represents the ceiling function. In other words, the arrival time of the l_N-th contact, $T_{if}^{I,l_N} = \sum_{l=1}^{l_N} t_{if}^{I_l}$, should be within the deadline T_D. Note that T_{if}^{I,l_N} is a superposition of l_N independent and identical exponential distributions with parameter λ_{if}. According to [28], T_{if}^{I,l_N} follows Gamma distribution with parameters l_N and λ_{if}, i.e., $T_{if}^{I,l_N} \sim \Gamma(l_N, \lambda_{if})$. Thus, we have

$$\begin{aligned}P(K_{if}=N) &= \int_0^{T_D} \frac{\lambda_{if}^{l_N} t^{l_N-1} e^{-\lambda_{if}t}}{\Gamma(l_N)} dt\\ &= \frac{\gamma(l_N, \lambda_{if}T_D)}{\Gamma(l_N)},\end{aligned} \quad (2)$$

where $\gamma(s,z) = \int_0^z t^{s-1}e^{-t}dt$ is the lower incomplete Gamma function and $\Gamma(n) = (n-1)!$ is the Gamma function.

With the probability of the number of successfully delivered file segments within T_D, the average file delivery delay for user i receiving the whole requested file f is expressed as

$$\mathcal{T}_{if} = \mathbb{E}(T_{if}) = \sum_{k=0}^{N} P(K_{if}=k)\mathbb{E}(T_{if}^k | K_{if}=k), \quad (3)$$

where T_{if}^k is the file delivery delay when k file segments are successfully transmitted within T_D. In fact, if the whole file f fails to be successfully transmitted within time T_D, i.e., $0 \le k < N$, the remaining segments should be delivered by BS. In this case, we have $T_{if}^k = T_D$. If file f is successfully shared within T_D, i.e., $K_{if}=N$, the file delivery delay is the summation of all l_N inter-contact time, i.e.,

$$T_{if}^N = \sum_{l=1}^{l_N} t_{if}^{I_l}. \quad (4)$$

According to the Poisson contact process, we have $T_{if}^N \sim \Gamma(l_N, \lambda_{if})$.

Therefore, we can derive that

$$\begin{aligned}\mathcal{T}_{if} &= \mathbb{E}(T_{if})\\ &= \sum_{k=0}^{N-1} P(K_{if}=k)T_D + \int_0^{T_D} t\frac{\lambda_{if}^{l_N} t^{l_N-1} e^{-\lambda_{if}t}}{\Gamma(l_N)} dt\\ &= \sum_{k=0}^{N-1} P(K_{if}=k)T_D + \frac{\gamma(l_N+1, \lambda_{if}T_D)}{\lambda_{if}\Gamma(l_N)}.\end{aligned} \quad (5)$$

Substituting (1) into (5), the detailed expression of the delay when user i receiving requested file f can be obtained. Consequently, the average file delivery delay for D2D networks is given as

$$\mathcal{T} = \mathbb{E}(T) = \frac{1}{U}\sum_{i=1}^{U}\sum_{f=1}^{F} p_{if}\mathcal{T}_{if}. \quad (6)$$

D. Problem Formulation

With the emerging of mobile video services, the user experience is more significant to improve the user viscosity. As an important factor impacting the user experience, the delay is gradually becoming a crucial wireless network performance. By bringing files much closer to users, D2D caching can reduce the file delivery delay and the delay performance heavily depends on the caching strategy of mobile users. Traditionally, the file popularity is mainly used in the cache placement design. As users in wireless networks moves all the time, leading to higher D2D communication chances, the user mobility may also be exploited in the cache placement design to reduce the network delay. Therefore, this paper studies mobility-aware caching strategies to reduce the average file delivery delay in D2D networks. Mathematically, the formulated delay-oriented cache placement problem considering the user mobility is given as

$$\mathcal{P}_C: \quad \min_{\boldsymbol{X}} \quad \mathcal{T} \quad (7a)$$

$$\text{s.t.} \quad \sum_{f=1}^{F} x_{if} \le \bar{C}_i, \forall i \in \mathcal{U}, \quad (7b)$$

$$x_{if} \in \{0,1\}, \forall i \in \mathcal{U}, \forall f \in \mathcal{F}, \quad (7c)$$

where $\bar{C}_i = C_i/S$ is the normalized cache capacity. (7b) is the cache capacity constraint and (7c) is the binary cache placement indicator constraint.

Notice that problem $\mathcal{P}_C$ is a NLIP problem with simple constraint. However, the objective function in problem $\mathcal{P}_C$ is very complicated, which makes it impossible to be handled by 0-1 relaxation and convex optimization. Hence, combinatorial optimization methods are adopted in this paper solve the discrete problem $\mathcal{P}_C$.

In what follows, the optimal caching strategy is obtained in Section III and a sub-optimal caching strategy with low complexity is achieved in Section IV.

III. DP-Based Optimal Mobility-Aware Caching Strategy

In this section, DP is adopted to handle the formulated problem $\mathcal{P}_C$ and an optimal caching strategy is proposed. The DP approach is used to effectively handle integer problems by breaking the original integer problem down into a multiple-stage decision problem, and sequentially solves decision problem in each stage. In particular, a recursive relationship between adjacent stages is derived and the optimal decision in each stage is made based on the optimal decision at previous stage. With the decision in the first stage and the recursive relationship, the optimal decision can be made stage-by-stage and finally the optimal solution to the original integer problem can be found. In what follows, we first reformulate problem $\mathcal{P}_C$ as a multi-stage decision problem, and then find the recursive relationship between adjacent stages.

A. Multi-Stage Decision Problem Reformulation

We divide problem $\mathcal{P}_C$ into F stages and optimize the cache placement for the first f files at the f-th stage. Denote $\bar{\boldsymbol{c}}_f = [\bar{c}_{if}]^{U\times 1}(0 \le \bar{c}_{if} \le \bar{C}_i)$ as the state vector at the f-th stage, representing cache capacities of all users for storing the first f files.

For state $\bar{\boldsymbol{c}}_f$ at the f-th stage, the cache placement problem for the first f files is expressed as

$$\mathcal{P}_{C_f}: \quad J_f(\bar{\boldsymbol{c}}_f) = \min_{\boldsymbol{x}_m, 1\le m\le f} \sum_{m=1}^{f} U_m(\boldsymbol{x}_m) \tag{8a}$$

$$\text{s.t.} \quad \sum_{m=1}^{f} x_{im} \le \bar{c}_{if}, \forall i \in \mathcal{U}, \tag{8b}$$

$$x_{im} \in \{0,1\},\ \forall i,\ 1 \le m \le f, \tag{8c}$$

where $\boldsymbol{x}_m = [x_{im}]^{U\times 1}$ is the cache placement vector for file m, and

$$U_m(\boldsymbol{x}_m) = \frac{1}{U}\sum_{i=1}^{U} p_{im}\mathcal{T}_{im}. \tag{9}$$

For the last stage decision problem with $f = F$ and $\bar{c}_{iF} = \bar{C}_i$, i.e., $\bar{\boldsymbol{c}}_F = [\bar{C}_i]^{U\times 1}$, problem $\mathcal{P}_{C_F}$ is expressed as

$$\mathcal{P}_{C_F}: \quad J_F(\bar{\boldsymbol{c}}_F) = \min_{\boldsymbol{x}_m, 1\le m\le F} \sum_{m=1}^{F}\frac{1}{U}\sum_{i=1}^{U} p_{im}\mathcal{T}_{im} \tag{10a}$$

$$\text{s.t.} \quad \sum_{m=1}^{F} x_{im} \le \bar{C}_i, \forall i \in \mathcal{U}, \tag{10b}$$

$$x_{im} \in \{0,1\},\ \forall i \in \mathcal{U},\ 1 \le m \le F. \tag{10c}$$

It is easy to observe that problem $\mathcal{P}_{C_F}$ is equivalent to the original problem $\mathcal{P}_C$. By finding the optimal decision of the first stage problem and establishing a recursive relationship between adjacent stage problems, the optimal decision of problem $\mathcal{P}_{C_f}$ can be achieved stage-by-stage. Finally, the original problem $\mathcal{P}_C$ is optimally solved.

Algorithm 1: DP-Based Optimal Mobility-Aware Caching Strategy.

1: Initialization
2: **for** all $\bar{\boldsymbol{c}}_1 \le [\bar{C}_i]^{U\times 1}$ **do**
3: $J_1(\bar{\boldsymbol{c}}_1) = U_1(\min\{\mathbf{1}, \bar{\boldsymbol{c}}_1\})$;
4: **end for**
5: **for** $f = 2$ to F **do**
6: **for** all $\bar{\boldsymbol{c}}_f \le [\bar{C}_i]^{U\times 1}$ **do**
7: Initialize $J_f(\bar{\boldsymbol{c}}_f)$ as a large positive constant;
8: **for** all possible $\boldsymbol{x}_f$ **do**
9: **if** $J_f(\bar{\boldsymbol{c}}_f) > U_f(\boldsymbol{x}_f) + J_{f-1}(\bar{\boldsymbol{c}}_{f-1})$ **then**
10: $J_f(\bar{\boldsymbol{c}}_f) = U_f(\boldsymbol{x}_f) + J_{f-1}(\bar{\boldsymbol{c}}_{f-1})$;
11: $\boldsymbol{x}_f^* = \boldsymbol{x}_f$;
12: **end if**
13: **end for**
14: **end for**
15: **end for**
16: **return** the cache placement matrix $\boldsymbol{X}^*$.

B. Recursive Relationship

At the first stage, the optimal decision of user i is to cache the first file if $\bar{c}_{i1} \ge 1$. Mathematically, $\boldsymbol{x}_1 = \min\{\mathbf{1}, \bar{\boldsymbol{c}}_1\}$. Thus, we have

$$J_1(\bar{\boldsymbol{c}}_1) = U_1(\min\{\mathbf{1}, \bar{\boldsymbol{c}}_1\}). \tag{11}$$

Then, at the subsequent f-th stage ($2 \le f \le F$), the optimal decision is made according to the decision at the $f-1$-th stage by using the following recursive relationship:

$$J_f(\bar{\boldsymbol{c}}_f) = \min_{\boldsymbol{x}_f} \ U_f(\boldsymbol{x}_f) + J_{f-1}(\bar{\boldsymbol{c}}_{f-1}) \tag{12a}$$

$$\text{s.t.} \quad x_{if} \in \{0,1\}, \forall i \in \mathcal{U}, \tag{12b}$$

$$x_{if} \le \bar{c}_{if},\ \forall i \in \mathcal{U}, \tag{12c}$$

where $\bar{\boldsymbol{c}}_{f-1} = \bar{\boldsymbol{c}}_f - \boldsymbol{x}_f$ is the state transition equation between adjacent stages.

Specifically, at the f-th stage with state $\bar{\boldsymbol{c}}_f$, the optimal decision of $\boldsymbol{x}_f$ can be achieved by exhaustively searching and finding the optimal one with the minimum value of $U_f(\boldsymbol{x}_f) + J_{f-1}(\bar{\boldsymbol{c}}_{f-1})$, where $J_{f-1}(\bar{\boldsymbol{c}}_{f-1})$ has been obtained at the $f-1$-th stage. Notice that the values of $J_{f-1}(\bar{\boldsymbol{c}}_{f-1})$ with all possible system states $\bar{\boldsymbol{c}}_{f-1}$ should be stored for the calling in the f-stage. Therefore, with $J_1(\bar{\boldsymbol{c}}_1)$ and the recursive relationship (12), problem $\mathcal{P}_C 1$ can be optimally solved. The optimal caching strategy is listed in Algorithm 1.

C. Complexity Analysis

According to Algorithm 1, the computational complexity of the stage 1 (i.e., line 2-line 4) is $\mathcal{O}((\bar{C}_{max}+1)^U)$, where $\bar{C}_{max} = \max \bar{C}_i$. For stage $f(1 < f \le F)$, the computational complexity (i.e., line 6-line 14) is $\mathcal{O}((2\bar{C}_{max}+1)^U)$. Hence,

the whole complexity is $\mathcal{O}((\bar{C}_{max}+1)^U)+\mathcal{O}(N_{ite}(F-1)(2\bar{C}_{max}+1)^U)$, i.e., $\mathcal{O}((F-1)(2\bar{C}_{max}+1)^U)$.

IV. Low-Complexity Greedy Caching Strategy

Although the DP-based caching strategy can obtain the optimal solution to problem $\mathcal{P}_C$, its complexity exponentially increases with respect to the number of mobile users in D2D networks. To lower the complexity, problem $\mathcal{P}_C$ is equivalently reformulated and a sub-optimal greedy caching strategy with linear complexity is then proposed in this section.

A. Problem Reformulation

Define $\mathcal{S}=\{x_{if}|i\in\mathcal{U}\text{ and }f\in\mathcal{F}\}$ as the ground set, and $\mathcal{S}_i=\{x_{if}|f\in\mathcal{F}\}$ as the set of all files that may be cached by user i. Denote by $\mathcal{X}$ the cache placement set and $x_{if}\in\mathcal{X}$ refers to that user i caches file f. Obviously, $\mathcal{X}$ is a subset of $\mathcal{S}$. With the finite ground set $\mathcal{S}$, we further define a partition matroid $\mathcal{M}=(\mathcal{S};\mathcal{Z})$ [29], where $\mathcal{Z}$ is expressed as

$$\mathcal{Z}=\{\mathcal{X}\subseteq\mathcal{S}:|\mathcal{X}\cap\mathcal{S}_i|\leq\bar{C}_i,\forall i\in\mathcal{U}\}. \tag{13}$$

It can be observed that the cache capacity constraints (7b) and (7c) in original problem $\mathcal{P}_C$ is equivalent to $\mathcal{X}\in\mathcal{Z}$.

Next, we rewrite the objective function in problem $\mathcal{P}_C$ as

$$\mathcal{T}(\mathcal{X})=\frac{1}{U}\sum_{i=1}^{U}\sum_{f=1}^{F}p_{if}\mathcal{T}_{if}(\mathcal{X})=\frac{1}{U}\sum_{i=1}^{U}\sum_{f=1}^{F}p_{if}t(\lambda_{if}(\mathcal{X})), \tag{14}$$

where

$$t(\lambda)=\sum_{k=0}^{N-1}P(K_{if}=k)T_D+\frac{\gamma(l_N+1,\lambda T_D)}{\lambda\Gamma(l_N)} \tag{15}$$

and

$$\lambda_{if}(\mathcal{X})=\sum_{j=1}^{U}\mu_{ij}\mathbb{1}_{\mathcal{X}}(x_{jf}) \tag{16}$$

with

$$\mathbb{1}_{\mathcal{X}}(x_{jf})=\begin{cases}1, & x_{if}\in\mathcal{X},\\ 0, & x_{if}\notin\mathcal{X}.\end{cases} \tag{17}$$

Consequently, problem $\mathcal{P}_C$ can be equivalently reformulated as

$$\mathcal{P}_C1:\ \max_{\mathcal{X}\in\mathcal{Z}}\ \widetilde{\mathcal{T}}(\mathcal{X}) \tag{18}$$

where $\widetilde{\mathcal{T}}(\mathcal{X})=-\mathcal{T}(\mathcal{X})$. In the sequel, the definition of the monotone submodular property is briefly shown, which is used to deal with problem $\mathcal{P}_C1$.

Definition: Denote by g a set function and $\mathcal{Y}$ a set with many elements with $\mathcal{Y}_1\subseteq\mathcal{Y}_2\subset\mathcal{Y}$.

- g is monotone if $g(\mathcal{Y}_2)-g(\mathcal{Y}_1)\geq 0$ with every $\mathcal{Y}_1\subseteq\mathcal{Y}_2$.
- g is submodular if for every element $y\in\mathcal{Y}-\mathcal{Y}_2$, we have $g(\mathcal{Y}_1\cup\{y\})-g(\mathcal{Y}_1)\geq g(\mathcal{Y}_2\cup\{y\})-g(\mathcal{Y}_2)$. In other words, g is submodular if its marginal gain, which is defined as $g(\mathcal{Y}_1\cup\{y\})-g(\mathcal{Y}_1)$, is non-increasing with the increased number of elements in $\mathcal{Y}_1$.

Algorithm 2: Greedy Mobility-Aware Caching Strategy.

1: Initialize $\mathcal{X}=\varnothing$, $\mathcal{X}^r=\mathcal{S}$;
2: Initialize $n:=1$;
3: **while** $|\mathcal{X}|<\sum_{i=1}^{U}\bar{C}_i$ **do**
4: $x_{if}^*=$ argmax$_{x_{if}\in\mathcal{X}^r}[\widetilde{\mathcal{T}}^{(n)}(\mathcal{X}\cup\{x_{if}\})-\widetilde{\mathcal{T}}^{(n)}(\mathcal{X})]$;
5: **if** $|\mathcal{X}\cap\mathcal{S}_i|=\bar{C}_i$ **then**
6: $\mathcal{X}^r=\mathcal{X}^r\setminus S_i$;
7: **end if**
8: Set $n:=n+1$;
9: **end while**
10: **return** $\mathcal{X}$.

Based on the definition above, the following two propositions are demonstrated.

Proposition 1: $t(\lambda)$ defined in (15) is non-increasing with λ and $\widetilde{\mathcal{T}}(\mathcal{X})$ is monotone non-decreasing over set $\mathcal{X}$.

Proof: Please see Appendix A. ■

Proposition 2: There exists a $\lambda_0\approx\frac{\ln((l_N+1)/2)}{T_D}>0$ such that, if $\lambda\geq\lambda_0$, $t(\lambda)$ is convex and $\widetilde{\mathcal{T}}(\mathcal{X})$ is submodular.

Proof: . Please see Appendix B. ■

Remark: Observing from Proposition 2, $\lambda\geq\frac{\ln((l_N+1)/2)}{T_D}$ is required to make $\widetilde{\mathcal{T}}(\mathcal{X})$ submodular. From the perspective of physical significance, λ is the effective contact rate between a user with requested file and other users caching the file in D2D caching networks. λT_D is the average number of user contacts within the deadline T_D. Accordingly, $\lambda T_D\geq\ln((l_N+1)/2)$ can be trivially satisfied in real world to offload data to D2D links, where l_N is the number of user contacts required to completely offload a file.

With Proposition 1 and Proposition 2, it is confirmed that $\widetilde{\mathcal{T}}(\mathcal{X})$ is monotone submodular. Recall that the constraint $\mathcal{X}\in\mathcal{Z}$ is a matroid constraint, problem $\mathcal{P}_C1$ is a monotone submodular function maximization problem over a matroid constraint. Accordingly, the greedy algorithm can be applied to solve problem $\mathcal{P}_C1$, which stores a file with maximal marginal gain in each iteration. It is worth pointing out that, with the property in problem $\mathcal{P}_C1$, this greedy algorithm has been proved to have $(1-1/e)$-optimality performance guarantee [30].

B. Greedy Mobility-Aware Caching Strategy

The proposed greedy mobility-aware caching strategy is outlined in Algorithm 2, where $\widetilde{\mathcal{T}}^{(n)}(\mathcal{X}\cup\{x_{if}\})-\widetilde{\mathcal{T}}^{(n)}(\mathcal{X})$ in line 4 is the marginal gain. As the core of the greedy caching strategy is to cache the file with the maximal marginal gain in each iteration, we have $x_{if}^*=\text{argmax}_{x_{if}\in\mathcal{X}^r}[\widetilde{\mathcal{T}}^{(n)}(\mathcal{X}\cup\{x_{if}\})-\widetilde{\mathcal{T}}^{(n)}(\mathcal{X})]$.

C. Complexity Analysis

In Algorithm 2, the complexity is $\mathcal{O}((\sum_{i=1}^{U}\bar{C}_i)UF)$, where $\sum_{i=1}^{U}\bar{C}_i$ is the number of loops in Algorithm 2 and $\mathcal{O}(UF)$ is the complexity in each loop. It should be noticed that, the complexity of greedy caching strategy is linear to U.

V. Simulation Results

Numerical simulations are conducted in this section to evaluate the performance of the proposed caching strategies. Note that both the proposed two caching strategies in this paper are implemented in a centralized way and taken charge of by BS. The optimized cache placement indicators are then distributed by BS to each user. In the simulation, $U = 20$ users in D2D networks are moving and contacting with each other randomly. The contact rate between different mobile users, $\mu_{ij}, \forall i \in \mathcal{U}, j \in \mathcal{U} \setminus \{i\}$, follows Gamma distribution $\Gamma(4.43, 1/1088)$ [22]. Within the deadline $T_D = 800s$, required files of mobile users are served by others caching the corresponding files via D2D communications. After the deadline T_D, unsuccessfully delivered file segments are directly served by BS. The number of files in the file library is $F = 50$ and each file size is $S = 50$ MB. To facilitate the transmission, each file is equally divided into $N = 5$ file segments and the number of file segments successfully delivered during each user contact is set as $N_c = 2$.

For simplicity, the storage of each mobile user is $C = C_i = 0.2\text{GB}, \forall i$, which can store 4 files at most. The file popularity of each mobile user follows Zipf distribution, i.e., $p_{if} = \frac{1f^{\beta_i}}{\sum_{m=1}^{F} 1m^{\beta_i}}, \forall f$, where β_i is a positive parameter, denoting the concentration of the file popularity. We set $\beta = \beta_i = 0.6$ in this section.

In the simulation, our proposed cache placement Algorithm 1 and Algorithm 2 are respectively labeled as '**Optimal mobility-aware caching**' and '**Greedy mobility-aware caching**'. For comparison, four other caching strategies are also considered.

- *Offloading-oriented caching:* This caching strategy is proposed in [23] to maximize the D2D offloading ratio considering the user mobility. The D2D offloading ratio is defined as the ratio of the average number of successfully delivered file segments via D2D communications to the total number of file segments.
- *Most popular caching:* The most popular files are stored by users depending on their corresponding file popularity until the cache capacity is full.
- *Zipf-based probabilistic caching:* The file request probability of each mobile user follows Zipf distribution. According to this probability, each mobile user randomly stores files. The higher file probability, the more likely to be stored by users.
- *Random caching:* In this caching strategy, no side information is used. Mobile users randomly cache all files based on the same probability.

A. The Effectiveness of the Proposed Greedy Mobility-Aware Caching Strategy

The effectiveness of the proposed greedy caching algorithm is demonstrated in this subsection. In particular, both the performance and the runtime of all caching algorithms are compared. As the computational complexity of DP-based optimal mobility-aware caching algorithm is high, a small scale network setting with $F = 20$ and $U = 6$ is considered in this part. Other network parameters are the same as before.

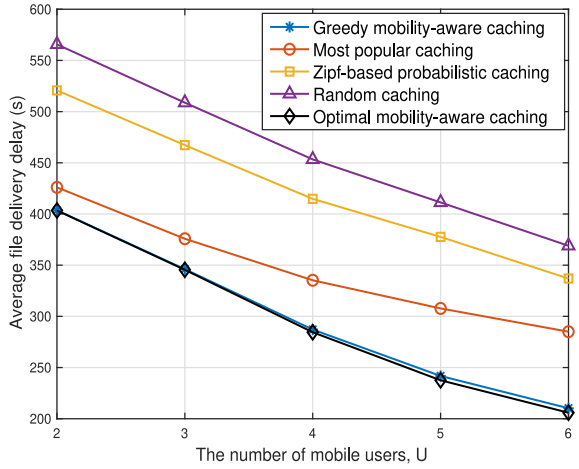


Fig. 3. Average file delivery delay versus the number of mobile users with $F = 20$, $C = 0.2$ GB, $\beta = 0.6$, $N = 5$ and $N_c = 2$.

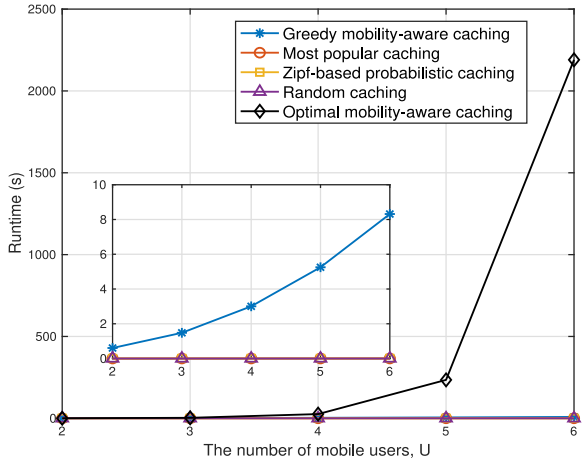


Fig. 4. The runtime of caching strategies versus the number of mobile user with $F = 20$, $C = 40$MB, $\beta = 0.6$, $N = 5$ and $N_c = 2$.

Fig. 3 shows how the number of mobile users affects the average file delivery delay for different caching strategies. With the increase of the number of mobile users, the average file delivery delay is decreasing. The reason is that, when the number of mobile users increases, the total cache capacity in the network becomes larger and mobile users have more chances to acquire interested files with D2D communications. Besides, both of the proposed optimal caching strategy and greedy caching strategy extremely outperform other caching strategies, owing to exploiting of the user mobility information. Moreover, the performance gap between two proposed caching strategies and the most popular caching strategy becomes larger as the number of mobile users increases. This is because the user mobility information is increasingly significant for the caching strategy design when the number of mobile users is larger. Most importantly, the proposed greedy mobility-aware caching strategy can achieve almost the same delay performance with the proposed optimal mobility-aware caching strategy.

Fig. 4 shows the runtime of all caching strategies with different number of mobile users. All simulations are run on a windows

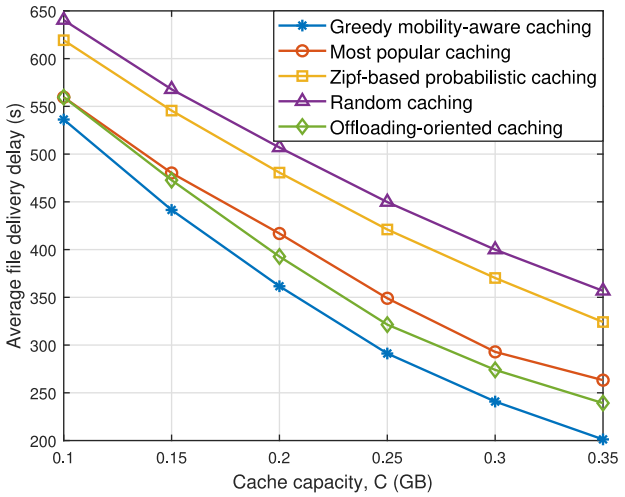


Fig. 5. Average file delivery delay versus cache capacity with $F = 50$, $U = 20$, $\beta = 0.6$, $N = 5$ and $N_c = 2$.

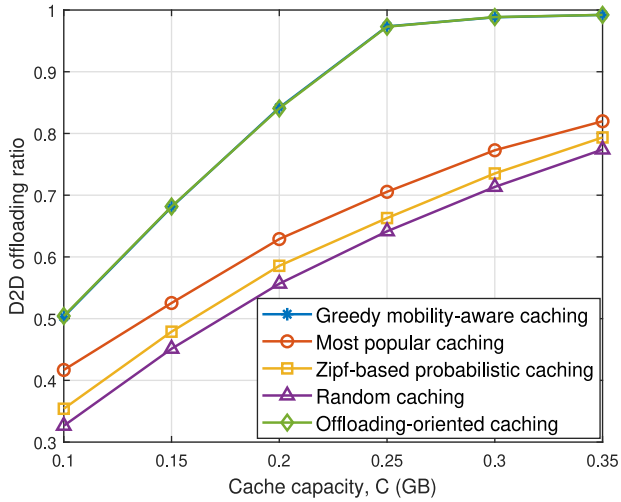


Fig. 6. D2D offloading ratio versus cache capacity with $F = 50$, $U = 20$, $\beta = 0.6$, $N = 5$ and $N_c = 2$.

server with 1.9 GHz Intel (R) Xeon (R) E5-2609 processor and 16 GB installed memory. It is straightforward that the runtime of the optimal mobility-aware caching strategy increases exponentially with the increase of U and that of the greedy mobility-aware caching strategy increases almost linearly with respect to U, which matches the theoretical computation complexity analysis in Section IV. Furthermore, it can be observed from Fig. 3 and Fig. 4 that, with the almost same delay performance, the runtime of the greedy mobility-aware caching strategy is dramatically shorter than that of the optimal mobility-aware caching strategy. For example, in Fig. 4, when $U = 6$, the runtime of the greedy mobility-aware caching strategy is 8.3s, while that of the optimal mobility-aware caching strategy' is 2190.2s being prohibitively long. This phenomenon shows the effectiveness of the proposed sub-optimal greedy mobility-aware caching strategy. Therefore, in what follows, only the proposed well-scalable greedy mobility-aware caching strategy is adopted for the performance comparison.

B. The Impact of Network Parameters on Caching Strategies

Fig. 5 shows how the cache capacity affects the delay performance of different caching strategies. In Fig. 5, the average file delivery delay is decreasing with the increase of the cache capacity. The reason is that, with enlarged capacity, more files will be stored by mobile users and then successfully delivered via D2D communications, resulting in lower file delivery delay. In addition, the proposed greedy mobility-aware caching strategy outperforms other caching strategies in terms of the average file delivery delay. This is due to the fact that the proposed delay-oriented caching strategy makes full use of the user mobility information.

Then, the impact of cache capacity on the D2D offloading ratio is also investigated in Fig. 6. As expected, the D2D offloading ratio of all caching strategies becomes higher with larger cache capacity. Moreover, the proposed greedy caching strategy achieves higher D2D offloading ratio than other caching strategies, except for the offloading-oriented caching strategy. Besides, when the cache capacity is small, with the increase of the cache capacity, the performance gap between the proposed greedy mobility-aware caching strategy and other caching strategies becomes wide. The reason is that, the user mobility information, fully exploited by the greedy mobility-aware caching strategy, gradually becomes significant in D2D caching. However, when the cache capacity is large enough, the performance gap becomes narrow. This is because that, for all caching strategies, almost all files can be cached by mobile users and then delivered via D2D communications. The narrowed performance gap is at the sacrifice of the cache capacity. Therefore, our proposed greedy mobility-aware caching strategy has significant performance advantage with the limited cache capacity, which is the common case in practice.

Observing Fig. 5 and Fig. 6, we can see that, with almost the same D2D offloading ratio, the proposed greedy mobility-aware caching strategy achieves lower average file delivery delay than the offloading-oriented caching strategy. From the definitions of these two metrics, we can find the reason. According to equation (5), the file delivery delay can be reduced when whole files are transmitted via D2D communications. Whereas, by the definition of the D2D offloading ratio, the D2D offloading ratio can be improved by transmitting more file segments. Note that a whole file being transmitted always means N file segments being delivered. By forcing more whole files to be delivered, the proposed greedy mobility-aware caching strategy can achieve almost the same D2D offloading ratio with the offloading-oriented caching strategy. Conversely, it is worth pointing out that N successfully transmitted file segments are not always from the same file. Hence, with more file segments being delivered, the offloading-oriented caching strategy can not achieve the same delay performance as the proposed greedy mobility-aware caching strategy.

Fig. 7 shows how the deadline, T_D, affects the average file delivery delay of different caching strategies. As expected, the proposed greedy mobility-aware caching strategy outperforms others in terms of the delay. In addition, when T_D enlarges, the

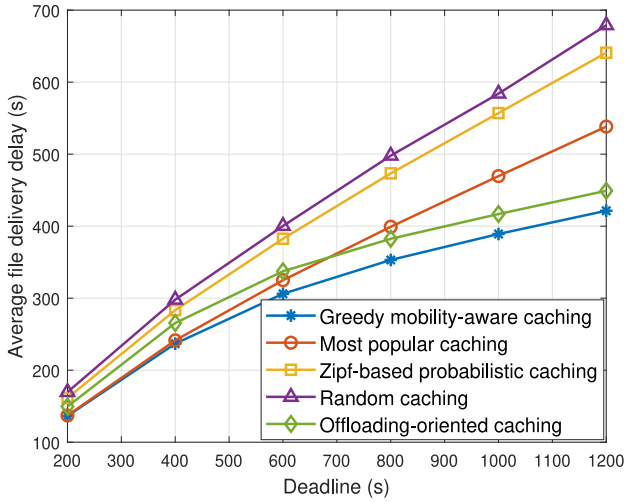


Fig. 7. Average file delivery delay versus the deadline with $F = 50$, $U = 20$, $C = 0.2$ GB, $\beta = 0.6$, $N = 5$ and $N_c = 2$.

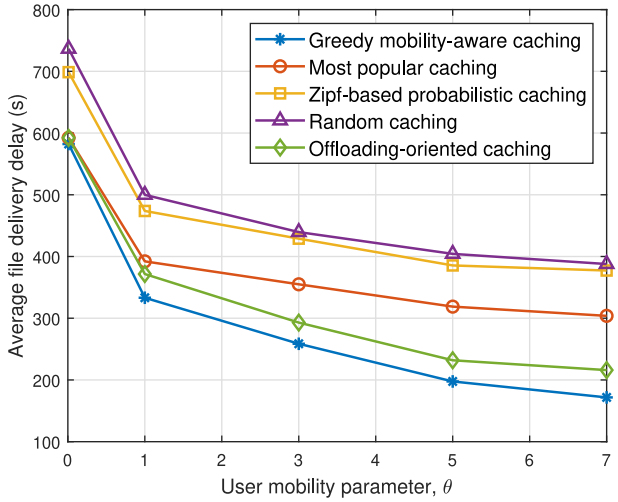


Fig. 9. Average file delivery delay versus the user mobility parameter with $F = 50$, $U = 20$, $C = 0.2$GB, $\beta = 0.6$, $N = 5$ and $N_c = 2$.

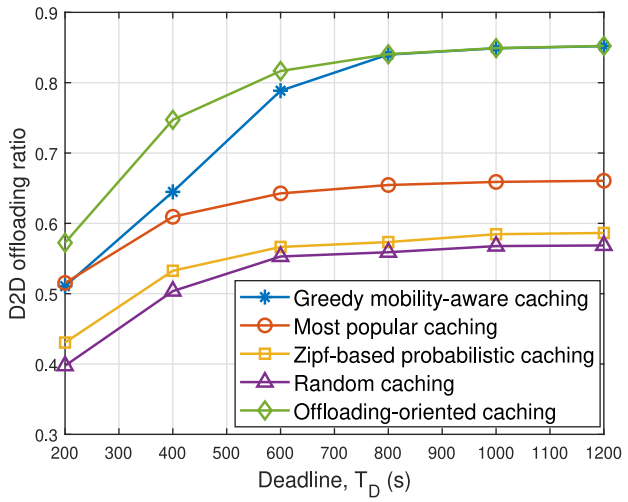


Fig. 8. D2D offloading ratio versus the deadline with $F = 50$, $U = 20$, $C = 0.2$ GB, $\beta = 0.6$, $N = 5$ and $N_c = 2$.

delay performance gap between the proposed greedy mobility-aware caching strategy and other caching strategies except for the offloading-oriented caching strategy is becoming wide. With larger T_D, mobile users have more time to move and share files via D2D communications, and the user mobility becomes more significant.

Fig. 8 shows how the deadline, T_D, affects the D2D offloading ratio for all caching strategies. As seen from Fig. 8, our proposed delay-oriented greedy mobility-aware caching strategy also outperforms the most popular caching strategy, the Zipf-based probabilistic caching strategy and the random caching strategy in terms of D2D offloading ratio. Besides, with the increase of T_D, the average file delivery delay in Fig. 7 is unfortunately increasing due to the long deadline setting, while the D2D offloading ratio in Fig. 8 is also increasing since mobile users have more time to contact and share files with each other. Therefore, the setting of the deadline should make a better tradeoff between the delay performance and the D2D offloading ratio.

Furthermore, it can be seen from Fig. 7 that, with all different values of the deadline, our proposed greedy mobility-aware caching strategy outperforms the offloading-oriented caching strategy in terms of the delay performance. However, for the D2D offloading ratio in Fig. 8, the offloading-oriented caching strategy achieves better performance than the proposed greedy mobility-aware caching strategy when T_D is small. This is because that, within small deadline T_D, the number of user contacts is not enough to transmit the whole file for most files and only parts of file segments from different files are delivered. Most importantly, the performance gap gradually becomes narrow with the increase of the deadline T_D. Especially, when T_D is more than 800s, these two caching strategies achieve almost the same D2D offloading ratio. This is due to the fact that more whole files are successfully transmitted with the increase of T_D.

C. The Impact of the User Mobility on Caching Strategies

Fig. 9 shows the influence of the user mobility on the average file delivery delay for different caching strategies. It has been demonstrated in [24] that the contact rate, μ_{ij}, is proportional to the user moving speed. As $\mu_{ij} \sim \Gamma(4.43, 1/1088)$, to model the different average user speed, we generate $\hat{\mu}_{ij} \sim \Gamma(4.43\theta^2, 1/(1088\theta))$. As a result, the average value of $\hat{\mu}_{ij}$ is θ times what μ_{ij} is while the variance remains the same, i.e., $\mathbb{E}(\hat{\mu}_{ij}) = \theta\mathbb{E}(\mu_{ij})$ and $Var(\hat{\mu}_{ij}) = Var(\mu_{ij})$. In Fig. 9, the average file delivery delays of all caching strategies decrease with the increase of θ. The reason is that, with larger θ, users have higher moving speed and more contact chances, which results in higher probability to obtain their request files via the low-delay D2D communications. Besides, with the increase of θ, the decrease trend of the delay is becoming slower. This is because that, when the user speed is large enough, each user may contact with almost all the other users in the network and the marginal gain becomes less. Furthermore, the proposed greedy

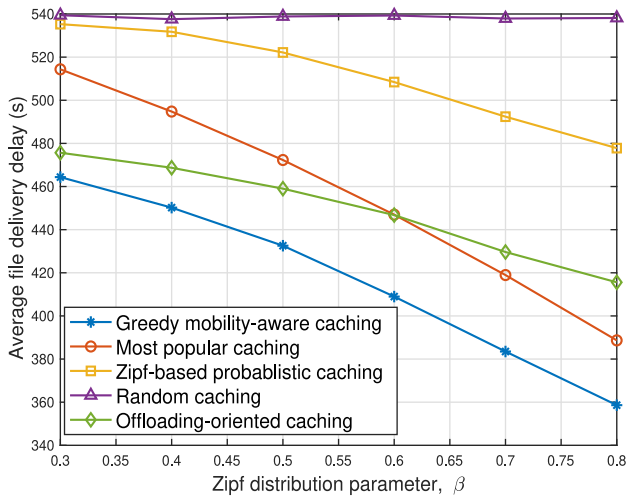


Fig. 10. Average file delivery delay versus Zipf distribution parameter, β, in real-world data set with $F = 50$, $U = 20$, $C = 0.2$ GB, $N = 5$ and $N_c = 2$.

caching strategy can achieve a much lower delay than other strategies. It can be observed that, when the proposed greedy mobility-aware caching strategy is adopted, the average delay can be reduced by 47% with $\theta = 7$, as compared with the most popular caching strategy. Accordingly, we can conclude that the user mobility information plays a key role in the caching strategy design, especially for the high mobility scenario.

D. The Performance of Caching Strategies in Real-World Data Set

In this paper, we also verify the effectiveness of our proposed caching strategy by the real-world user contact data set recorded in CRAWDAD [31]. The data set is collected in INFOCOM 2005 conference lasting for 4 days, where users equipped with IMotes are moving randomly in the conference and their contact data is recorded by the IMote, a bluetooth device. In our simulation, we first use the data of the second daytime to estimate the contact rate between user pairs, i.e., μ_{ij}, and choose 20 most active users. With the estimated contact rate, our proposed delay-oriented greedy mobility-aware caching strategy and other caching strategies are then designed. Finally, the performance of different caching strategies shown in Fig. 10 is obtained using the data of the third daytime. It is straightforward that the average file delivery delays of all caching strategies except for the random caching strategy are decreasing with the increase of the parameter in Zipf distribution, β. With the higher β, the file popularity is more concentrated and user requests are more likely to be satisfied by nearby users, resulting in low delivery delay. As expected, the proposed greedy mobility-aware caching strategy extremely outperforms others caching strategies, which effectively shows the advantage of our proposed greedy mobility-aware caching strategy.

VI. Conclusion

In this paper, we have investigated the delay-oriented cache placement problem considering the user mobility in D2D networks. With the inter-contact mobility model, the average file delivery delay has been analytically obtained. Then, a delay minimization cache placement problem considering the user mobility has been formulated with the limited cache capacity. Both DP-based optimal mobility-aware caching strategy and sub-optimal greedy mobility-aware caching strategy have been proposed. In the simulation, with the almost same delay performance, the proposed greedy mobility-aware caching strategy consumes remarkably shorter runtime than the optimal mobility-aware caching strategy, which validates the time-efficiency of the proposed greedy caching strategy. Furthermore, by making full use of the user mobility, the proposed mobility-aware caching strategy has advantage over other caching strategies in terms of the delay performance. This is verified by both simulated environment and the real data set.

Appendix A

As the feature of the delay function in (14) is completely different from that of the D2D offloading ratio function in the literature [23], [24], the proof method in [23], [24] could not be applied. Thus, a proof method exploiting the structure of the delay function is proposed in this Appendix. That is, the monotony of $\widetilde{\mathcal{T}}(\mathcal{X})$ is proved by demonstrating that $t(\lambda)$ is non-increasing with respect to λ.

We first prove that $t(\lambda)$ defined in (15) is a non-increasing function with respect to λ. Let $t_1(\lambda) = \sum_{k=0}^{N-1} P(K_{if} = k) T_D$ and $t_2(\lambda) = \frac{\gamma(l_N+1, \lambda T_D)}{\lambda \Gamma(l_N)}$.

Before tackling $t_1(\lambda)$, we first reformulate it as a tractable form. Observing from (1) that, when file f is not completely transmitted within deadline T_D (i.e., $0 \le k < N$), $P(K_{if} = k) = 0$ if $k \mod N_c \neq 0$. By removing these trivial cases, $t_1(\lambda)$ can be equivalently rewritten as

$$t_1(\lambda) = \sum_{l=0}^{l_N-1} \frac{e^{-\lambda T_D} (\lambda T_D)^l}{l!} T_D. \tag{19}$$

$l_N - 1$ in (19) is because that there are at most $l_N - 1$ user contacts within T_D with $0 \le k < N$. To analyze the monotonicity of $t_1(\lambda)$, the first derivative of $t_1(\lambda)$ is expressed as

$$\begin{aligned} t_1'(\lambda) &= e^{-\lambda T_D} \left(- \sum_{l=0}^{l_N-1} \frac{T_D^{l+2}}{l!} \lambda^l + \sum_{l=1}^{l_N-1} \frac{T_D^{l+1}}{(l-1)!} \lambda^{l-1} \right) \\ &= e^{-\lambda T_D} \left(- \sum_{l=0}^{l_N-1} \frac{T_D^{l+2}}{l!} \lambda^l + \sum_{n=0}^{l_N-2} \frac{T_D^{n+2}}{n!} \lambda^n \right) \\ &= e^{-\lambda T_D} \left(- \frac{T_D^{l_N+1}}{(l_N-1)!} \lambda^{l_N-1} \right). \end{aligned} \tag{20}$$

Then, for the derivative of $t_2(\lambda)$, the greatest challenge is the handling of the complicated lower incomplete gamma function $\gamma(l_N + 1, \lambda T_D)$. To tackle this obstacle, an equation represented as (21) is adopted to rewrite the lower incomplete gamma

function [32].

$$\gamma(s,z) = \int_0^z t^{s-1}e^{-t}dt = z^s\Gamma(s)e^{-z}\sum_{k=0}^{\infty}\frac{z^k}{\Gamma(s+k+1)}. \quad (21)$$

This equivalent reformulation is crucial for the following derivation. According to (21), $t_2(\lambda)$ can be equivalently reformulated as

$$t_2(\lambda) = e^{-\lambda T_D}\sum_{k=0}^{\infty}\frac{l_N}{\Gamma(l_N+k+2)}T_D^{l_N+k+1}\lambda^{l_N+k}. \quad (22)$$

Taking the derivative of λ, we can obtain

$$\begin{aligned}
&t_2'(\lambda)\\
&= e^{-\lambda T_D}\left(-\sum_{k=0}^{\infty}\frac{l_N}{\Gamma(l_N+k+2)}T_D^{l_N+k+2}\lambda^{l_N+k}\right.\\
&\quad\left.+\sum_{k=0}^{\infty}\frac{l_N(l_N+k)}{\Gamma(l_N+k+2)}T_D^{l_N+k+1}\lambda^{l_N+k-1}\right)\\
&= e^{-\lambda T_D}\left(-\sum_{n=1}^{\infty}\frac{l_N}{\Gamma(l_N+n+1)}T_D^{l_N+n+1}\lambda^{l_N+n-1}\right.\\
&\quad\left.+\sum_{k=0}^{\infty}\frac{l_N(l_N+k)}{\Gamma(l_N+k+2)}T_D^{l_N+k+1}\lambda^{l_N+k-1}\right)\\
&= e^{-\lambda T_D}\left(-\sum_{k=1}^{\infty}\frac{l_N}{(l_N+k+1)!}T^{l_N+k+1}\lambda^{l_N+k-1}\right.\\
&\quad\left.+\frac{l_N^2}{(l_N+1)!}T_D^{l_N+1}\lambda^{l_N-1}\right). \quad (23)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&t'(\lambda)\\
&= t_1'(\lambda)+t_2'(\lambda)\\
&= e^{-\lambda T_D}\left(\left(\frac{l_N^2}{(l_N+1)!}-\frac{1}{(l_N-1)!}\right)T_D^{l_N+1}\lambda^{l_N-1}\right.\\
&\quad\left.-\sum_{k=1}^{\infty}\frac{l_N}{(l_N+k+1)!}T_D^{l_N+k+1}\lambda^{l_N+k-1}\right)\\
&= e^{-\lambda T_D}\left(-\frac{l_N!}{(l_N+1)!(l_N-1)!}T_D^{l_N+1}\lambda^{l_N-1}\right.\\
&\quad\left.-\sum_{k=1}^{\infty}\frac{l_N}{(l_N+k+1)!}T_D^{l_N+k+1}\lambda^{l_N+k-1}\right)\\
&= -e^{-\lambda T_D}\left(\sum_{k=0}^{\infty}\frac{l_N}{(l_N+k+1)!}T_D^{l_N+k+1}\lambda^{l_N+k-1}\right)\\
&\le 0. \quad (24)
\end{aligned}$$

This proves that $t(\lambda)$ is non-increasing.

Finally, we demonstrate that $\widetilde{\mathcal{T}}(\mathcal{X})$ is monotone non-decreasing. Let $\mathcal{Y}\subseteq\mathcal{S}$ and $x_{jf}\in\mathcal{S}-\mathcal{Y}$, due to the linearity of $\lambda_{if}(\mathcal{X})$, we have $\lambda_{if}(\mathcal{Y}\cup\{x_{jf}\})-\lambda_{if}(\mathcal{Y})\ge 0$. Since $t(\lambda)$ is non-increasing, $t(\lambda_{if}(\mathcal{Y}\cup\{x_{jf}\}))-t(\lambda_{if}(\mathcal{Y}))\le 0$, which results in the non-increasing $t(\lambda_{if}(\mathcal{X}))$. Owing to the fact that $\mathcal{T}(\mathcal{X})$ is non-negative linear combination of $t(\lambda_{if}(\mathcal{X}))$ and $t(\lambda_{if}(\mathcal{X}))$ is monotone non-increasing, $\mathcal{T}(\mathcal{X})$ is also monotone non-increasing. Hence, $\widetilde{\mathcal{T}}_{if}(\mathcal{X}) = -\mathcal{T}_{if}(\mathcal{X})$ is monotone non-decreasing over set $\mathcal{X}$. ■

APPENDIX B

In this Appendix, we first demonstrate that $t(\lambda)$ is convex when $\lambda\ge\lambda_0$. Taking the derivative of $t'(\lambda)$ with respect to λ, with similar tricks as (23), we have

$$t''(\lambda) = e^{-\lambda T_D}T_D^{l_N+1}\lambda^{l_N-2}l_N\times\left(\sum_{k=0}^{\infty}\frac{2}{(l_N+k+2)!}T_D^{k+1}\lambda^{k+1}-\frac{l_N-1}{(l_N+1)!}\right). \quad (25)$$

Let $h(\lambda)=\sum_{k=0}^{\infty}\frac{2}{(l_N+k+2)!}T_D^{k+1}\lambda^{k+1}$, it can be observed that $h(\lambda)$ is an increasing function. In addition, $h(\lambda)$ can be rewritten as

$$h(\lambda) = \frac{2}{(\lambda T_D)^{l_N+1}}\left(e^{\lambda T_D}-\sum_{k=0}^{l_N+1}\frac{(\lambda T_D)^k}{k!}\right), \quad (26)$$

which is derived from the following equation

$$\begin{aligned}
&(\lambda T_D)^{l_N+1}h(\lambda)\\
&= \sum_{k=0}^{\infty}\frac{2}{(l_N+k+2)!}(\lambda T_D)^{l_N+k+2}\\
&= 2\sum_{k=l_N+2}^{\infty}\frac{(\lambda T_D)^k}{k!}\\
&= 2\left(e^{\lambda T_D}-\sum_{k=0}^{l_N+1}\frac{(\lambda T_D)^k}{k!}\right). \quad (27)
\end{aligned}$$

Then, applying L'Hospital's rule $N+1$ times to $h(\lambda)$, we have

$$\begin{aligned}
&\lim_{\lambda\to 0}h(\lambda)\\
&= \lim_{\lambda\to 0}\frac{2\left(e^{\lambda T_D}-\sum_{k=0}^{l_N+1}\frac{(\lambda T_D)^k}{k!}\right)}{(\lambda T_D)^{l_N+1}}\\
&= \lim_{\lambda\to 0}\frac{2(e^{\lambda T_D}-1)}{(l_N+1)!}\\
&= 0. \quad (28)
\end{aligned}$$

Similarly, $\lim_{\lambda\to\infty}h(\lambda)=\infty$. Recall that $h(\lambda)$ is an increasing function with λ, there must be a $\lambda_0>0$ such that $h(\lambda_0)-(l_N-1)/(l_N+1)!\ge 0$ when $\lambda\ge\lambda_0$. Therefore, if $\lambda>\lambda_0$, $t''(\lambda)=h(\lambda_0)-(l_N-1)/(l_N+1)!\ge 0$. This completes the proof that $t(\lambda)$ is convex when $\lambda\ge\lambda_0$.

When $\lambda \to 0$, $h(\lambda)$ can be approximated as $\frac{2(e^{\lambda T_D}-1)}{(l_N+1)!}$ according to (28). Let $\frac{2(e^{\lambda T_D}-1)}{(l_N+1)!} \geq \frac{l_N-1}{(l_N+1)!}$, we can derive that, when $\lambda \geq \frac{\ln((l_N+1)/2)}{T_D}$, $t''(\lambda) \geq 0$. Hence, we have $\lambda_0 \approx \frac{\ln((l_N+1)/2)}{T_D}$.

Finally, we show that $\widetilde{\mathcal{T}}(\mathcal{X})$ is submodular. Since $t(\lambda)$ is decreasing convex when $\lambda > \lambda_0$ and $\lambda_{if}(\mathcal{X})$ is monotone increasing, $\mathcal{T}_{if}(\mathcal{X})$ is decreasing and the decrease trend is becoming slower. Thus, $\widetilde{\mathcal{T}}_{if}(\mathcal{X})$ is an increasing set function with gradual slower increasing trend.

Let $\mathcal{Y}_1 \subset \mathcal{Y}_2 \subset \mathcal{S}$, $x_{jf} \in \mathcal{S} - \mathcal{Y}_2$ and define

$$\Delta_{\widetilde{\mathcal{T}}}(\mathcal{Y}) = \widetilde{\mathcal{T}}_{if}(\mathcal{Y} \cup \{x_{jf}\}) - \widetilde{\mathcal{T}}_{if}(\mathcal{Y}), \tag{29}$$

we have

$$\begin{aligned}\Delta_{\widetilde{\mathcal{T}}}(\mathcal{Y}_1) - \Delta_{\widetilde{\mathcal{T}}}(\mathcal{Y}_2) &= \widetilde{\mathcal{T}}_{if}(\mathcal{Y}_1 \cup \{x_{jf}\}) - \widetilde{\mathcal{T}}_{if}(\mathcal{Y}_1) \\ &- \left(\widetilde{\mathcal{T}}_{if}(\mathcal{Y}_2 \cup \{x_{jf}\}) - \widetilde{\mathcal{T}}_{if}(\mathcal{Y}_2)\right) \geq 0. \end{aligned} \tag{30}$$

Therefore, according to the definition in Section IV, $\widetilde{\mathcal{T}}(\mathcal{X})$ is a submodular set function when $\lambda > \lambda_0$. ■

REFERENCES

[1] I. Cisco Visual Networking, "Global mobile data traffic forecast update, 2017–2022 white paper," San Jose, CA, USA, Cisco Systems, Inc., White Paper, Feb. 2019. [Online]. Available: http://goo.gl/ylTuVx

[2] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 80–87, Jun. 2018.

[3] L. Lyu *et al.*, "Dynamics-aware and beamforming-assisted transmission for wireless control scheduling," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7767–7690, Nov. 2018.

[4] R. Sun *et al.*, "QoE-Driven transmission-aware cache placement and cooperative beamforming design in Cloud-RANs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 636–650, Jan. 2020.

[5] 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 Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[6] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1791–1805, Aug. 2018.

[7] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-device communication in LTE-advanced networks: A survey," *IEEE Commun. Surv. Tut.*, vol. 17, no. 4, pp. 1923–1940, Dec. 2015.

[8] H. Nishiyama, M. Ito, and N. Kato, "Relay-by-smartphone: Realizing multihop device-to-device communications," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 56–65, Apr. 2014.

[9] 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 Commun.*, vol. 23, no. 4, pp. 74–81, Aug. 2016.

[10] L. Wang, H. Wu, Y. Ding, W. Chen, and H. V. Poor, "Hypergraph-based wireless distributed storage optimization for cellular D2D underlays," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2650–2666, Oct. 2016.

[11] W. Chen and H. V. Poor, "Content pushing with request delay information," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1146–1161, Mar. 2017.

[12] Y. Lu, W. Chen, and H. V. Poor, "Content pushing based on physical layer multicasting and request delay information," in *Proc. IEEE Global Commun. Conf.*, Dec. 2016, pp. 1–6.

[13] Y. Li, M. C. Gursoy, and S. Velipasalar, "A delay-aware caching algorithm for wireless D2D caching networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, May 2017, pp. 456–461.

[14] N. Deng and M. Haenggi, "The benefits of hybrid caching in gauss–poisson D2D networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1217–1230, Jun. 2018.

[15] D. Malak, M. Al-Shalash, and J. G. Andrews, "Optimizing content caching to maximize the density of successful receptions in device-to-device networking," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4365–4380, Oct. 2016.

[16] Y. Guo, L. Duan, and R. Zhang, "Cooperative local caching under heterogeneous file preferences," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 444–457, Jan. 2017.

[17] N. Cheng, H. Zhou, L. Lei, N. Zhang, Y. Zhou, X. Shen, and F. Bai, "Performance analysis of vehicular device-to-device underlay communication," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5409–5421, Jun. 2017.

[18] H. Peng, D. Li, Q. Ye, K. Abboud, H. Zhao, W. Zhuang, and X. Shen, "Resource allocation for cellular-based inter-vehicle communications in autonomous multiplatoons," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11 249–11 263, Dec. 2017.

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

[20] J. Qiao, Y. He, and X. Shen, "Proactive caching for mobile video streaming in millimeter wave 5G networks." *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 7187–7198, Oct. 2016.

[21] S. Zhang, W. Quan, J. Li, W. Shi, P. Yang, and X. Shen, "Air-ground integrated vehicular network slicing with content pushing and caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2114–2127, Sep. 2018.

[22] A. Passarella and M. Conti, "Analysis of individual pair and aggregate intercontact times in heterogeneous opportunistic networks," *IEEE Trans. Mob. Comput.*, vol. 12, no. 12, pp. 2483–2495, Oct. 2013.

[23] R. Wang, J. Zhang, S. Song, and K. B. Letaief, "Mobility-aware caching in D2D networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5001–5015, Aug. 2017.

[24] R. Wang, J. Zhang, S. Song, and K. B. Letaief, "Exploiting mobility in cache-assisted D2D networks: Performance analysis and optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5529–5605, Oct. 2018.

[25] M. Chen, Y. Hao, L. Hu, K. Huang, and V. K. Lau, "Green and mobility-aware caching in 5G networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 8347–8361, Dec. 2017.

[26] T. Deng, G. Ahani, P. Fan, and D. Yuan, "Cost-optimal caching for D2D networks with user mobility: Modeling, analysis, and computational approaches," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3082–3094, May 2018.

[27] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE Conf. Comput. Commun.*, Mar. 1999, pp. 126–134.

[28] S. M. Ross, *Introduction to Probability Models*. New York, NY, USA: Academic, 2014.

[29] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.

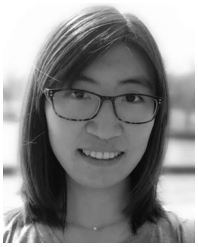
[30] M. Conforti and G. Cornuéjols, "Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem," *Discrete Appl. Math.*, vol. 7, no. 3, pp. 251–274, Mar. 1984.

[31] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD dataset cambridge/haggle (v. 2009-05-29)," May 2009. [Online]. Available: https://crawdad.org/cambridge/haggle/20090529

[32] F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, *NIST Handbook of Mathematical Functions Hardback and CD-ROM*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

Ruijin Sun (Student Member, IEEE) received the B.S. degree from the Chongqing University of Posts and Telecommunications, China, in 2013, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, China, in 2019. She is currently working as a joint Postdoctoral Fellow with Peng Cheng Laboratory and Tsinghua University. She was a Visiting Student at the University of Waterloo, Canada (September 2017–September 2018). Her research interests are in the area of MIMO and wireless caching networks.

Ying Wang (Member, IEEE) received the Ph.D. degree in circuits and systems from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2003. She is currently a Professor with BUPT and the Director of the Radio Resource Management Laboratory, Wireless Technology Innovation Institute, BUPT. Her research interests are in the area of the cooperative systems and radio resource management in wireless communications. She is active in standardization activities of 3GPP and ITU. She took part in performance evaluation work of the Chinese Evaluation Group, as a Representative of BUPT. She was a recipient of first prizes of the Scientific and Technological Progress Award by the China Institute of Communications in 2006 and 2009, respectively, and a second prize of the National Scientific and Technological Progress Award in 2008. She was also selected in the New Star Program of Beijing Science and Technology Committee and the New Century Excellent Talents in University, Ministry of Education, in 2007 and 2009, respectively. She has authored over 100 papers in international journals and conferences proceedings.



Ling Lyu (Student Member, IEEE) received the B.S. degree in telecommunication engineering from Jinlin University, Changchun, P. R. China, in 2013; and the Ph.D. degree in control theory and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2019. She joined Dalian Maritime University, China, in 2019, where she is currently an Associate Professor with the Department of Telecommunication. She was a visiting student in University of Waterloo, Canada (September 2017–September 2018). Her current research interests include wireless sensor and actuator network and application in industrial automation, the joint design of communication and control in industrial cyber-physical systems, estimation and control over lossy wireless networks, machine type communication enabled reliable transmission in the fifth generation network, resource allocation, energy efficiency.

Nan Cheng (Member, IEEE) received the Ph.D. degree from the University of Waterloo in 2016, and B.E. degree and the M.S. degree from Tongji University, Shanghai, China, in 2009 and 2012, respectively. He is currently a Professor with the School of Telecommunication Engineering, and with State Key Lab of ISN, Xidian University, Shaanxi, China. He worked as a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, from 2017 to 2018. His current research focuses on space-air-ground integrated system, big data in vehicular networks, and self-driving system. His research interests also include performance analysis, MAC, opportunistic communication, and application of AI for vehicular networks. He is also interested in space-air-ground integrated networks.

Shan Zhang (Member, IEEE) received the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2016. She is currently an Assistant Professor at the School of Computer Science and Engineering, Beihang University, Beijing, China. She was a Postdoctoral Fellow with the Department of Electronical and Computer Engineering, University of Waterloo, Ontario, Canada, from 2016 to 2017. Her research interests include mobile edge computing, wireless network virtualization and intelligent management. She received the Best Paper Award at the Asia-Pacific Conference on Communication in 2013.

Tingting Yang (Member, IEEE) received the B.Sc. and Ph.D. degrees from Dalian Maritime University, China, in 2004 and 2010, respectively. She is currently a Professor at the School of Electrical Engineering and Intelligentization, Dongguan University of Technology, China. From September 2012 to August 2013, she was Visiting Scholar at the Broadband Communications Research (BBCR) Lab at the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Her research interests are in the areas of maritime wideband communication networks, DTN networks, green wireless communications. She serves as the Associate Editor-in-Chief of the IET Communications, as well as the Advisory Editor for SpringerPlus. She also served as a TPC Member for IEEE ICC'14 and ICC'15 Conference.

Xuemin (Sherman) Shen (Fellow, IEEE) received the B.Sc. degree from Dalian Maritime University, China, in 1982, and the M.Sc. and Ph.D. degrees from Rutgers University, New Jersey, USA, in 1987 and 1990, respectively, all in electrical engineering. He is a University Professor and the Associate Chair for Graduate Studies, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on resource management, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He served as the Technical Program Committee Chair/Co- Chair for IEEE Globecom'16, Infocom'14, IEEE VTC'10 Fall, and Globecom'07, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the General Co-Chair for ACM Mobihoc'15, Chinacom'07 and the Chair for IEEE Communications Society Technical Committee on Wireless Communications. He also serves/served as the Editor-in-Chief for IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, PEER-TO-PEER NETWORKING AND APPLICATION, and IET COMMUNICATIONS; a Founding Area Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS; an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, COMPUTER NETWORKS, AND ACM/WIRELESS NETWORKS, etc., and the Guest Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, and *IEEE Communications Magazine*, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.