Noncoordinated Individual Preference Aware Caching Policy in Wireless D2D Networks

Ming-Chun Lee, *Student Member, IEEE*, and Andreas F. Molisch, *Fellow, IEEE*

Ming Hsieh Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA, USA
Email: mingchul@usc.edu; molisch@usc.edu

***Abstract*—Recent investigations showed that cache-aided device-to-device (D2D) networks can be improved by properly exploiting the individual preferences of users. Since in practice it might be difficult to make centralized decisions about the caching distributions, this paper investigates the individual preference aware caching policy that can be implemented distributedly by users without coordination. The proposed policy is based on categorizing different users into different reference groups associated with different caching policies according to their preferences. To construct reference groups, learning-based approaches are used. To design caching policies that maximize throughput and hit-rate, optimization problems are formulated and solved. Numerical results based on measured individual preferences show that our design is effective and exploiting individual preferences is beneficial.**

I. Introduction

The rapidly increasing demand for videos presents a significant challenge for next-generation wireless networks. Exploiting the high concentration of video requests on popular files and availability of cheap storage, caching of video files at the wireless edge emerged as a promising solution, and has been widely discussed in the past years [1], [2].

Among the most popular implementations of this principle, cache-aided device-to-device (D2D) networks combine high-spectrum-efficiency D2D communications with on-device caching in wireless networks [1]–[4]. Both analytical and empirical results showed that the cache-aided D2D can effectively convert memory into bandwidth [4] and outperform conventional unicasting from base stations (BSs) [3], [4].

Though various aspects of cache-aided D2D networks were widely studied [4]–[8], most of the existing papers assume a homogeneous preference model, in which users in the network send requests following the same global popularity distribution. This modeling choice is mainly due to the better tractability of this model and the lack of practical individual (user) preference modeling. However, papers based on this model might be restricted since both intuition and measurements show that different users indeed have different preferences [9], [10]. Therefore, cache-aided D2D networks considering the more flexible heterogeneous preference modeling, in which users can have different requesting and caching behaviors, recently start to draw attentions [9]–[14].

Several papers have shown that considering heterogeneous preference modeling could help improving the network performance [11]–[14]. By assuming users in different groups have different preferences, [11] investigated the hit-rate optimization without fully considering the individual preferences. Exploiting knowledge of individual preferences, [12] proposed a deterministic caching policy to maximize the offloading probability. Based on the estimated user preferences, [13] proposed a centralized approach to determine the cached files for delay minimization. Most recently, under the assumption that users perfectly know individual preferences of one another, [14] proposed a caching policy design framework that can optimize throughput, energy efficiency, and hit-rate, and demonstrated based on simulations with a practical setup that cache-aided D2D networks can be significantly improved when the information of individual preferences is properly used.

In spite of this recent progress, the development of the caching policy design being aware of individual preferences is still at an early stage. Especially, to our best knowledge, most investigations consider a centralized caching policy and/or assume users can coordinate when caching files. However, since users, in general, are mobile and might not know at the time at which they cache the files (typically during night) who their neighbors will be at the time of file exchange (during the day), a design approach assuming centralized/cooperative caching between users might not always be feasible. In contrast, a design approach where users can decide their caching policies without knowing the individual preferences and caching policies of other users can be easily implemented in general situations. Therefore having such a noncoordinated caching policy design is of great interest.

In this paper, we propose a noncoordinated individual preference aware caching policy for optimizing throughput and hit-rate in BS-assisted wireless D2D caching networks adopting clustering and random-push scheduling [7]. Such policy is implemented based on a group-wise structure, in which there are reference groups and each group is associated with a unique probabilistic caching policy. Then as those reference

This work was supported in part by the National Science Foundation (NSF) under CNS-1816699 and CCF-1423140.

groups are designed in advance and users can be assigned to different reference groups according to their preferences, users can determine their caching policies without any coordination among users.

To optimize the proposed caching policy, we first propose a group assigning approach for users, and then optimize the reference groups and their corresponding caching policies. The optimization of the reference groups is conducted by combining the Kullback-Leibler (KL) divergence with the hierarchical and K-mean clustering algorithms [15] with the aid of user samples generated from the measurement based preference model [9], [10]. Based on the grouping result, we analyze the network and formulate the caching policy design problem aiming for throughput and hit-rate optimizations. An iterative algorithm is proposed to solve these two problems. Using the practical individual preferences generating by models and parameterizations in [9] and [10], respectively, we evaluate our design and provide comparisons with reference designs. Our results demonstrate the efficacy of the proposed design.

II. Network Model

In this paper, we consider a BS-assisted wireless D2D caching network, where each user can obtain the desired files through their own caches, caches of neighboring users via D2D links, or BS links. Each user device can cache S files. Users can be active or inactive. An active user is a user who has a request to be satisfied and participates in the D2D cooperation, i.e., sends files to other users if requested; an inactive user is a user who does not have a request but still participates in the D2D cooperation. If a user neither has a request nor participates in the D2D cooperation, such user is independent to the D2D network, and is neglected without loss of generality. We consider in the following only a single BS; this is no restriction of generality as long as inter-cell interference is limited by conventional means to allow supplying all users in the cell with a fixed required data rate (see below).

We consider the clustering network model [7]. A cell is served by a BS, and is split into different equally-sized square clusters with side length D. Users can cooperate via D2D links only with users in the same cluster. To avoid intra- and inter-cluster interference, we assume each cluster to have at most one D2D link and let different clusters use different time/frequency resources via "spatial reuse". The side length D is thereafter called "cluster size" or "cooperation distance". We assume a D2D communication in a cluster can always be successful, and occurs at a fixed rate, if the link is established. Such assumption is achievable by having appropriate system-level power control and frequency diversity (see details in Sec. II.A of [7]). Similarly, we assume that a BS communication (using a fixed rate) can always be successful if a BS link is scheduled. Users are distributed in the cell following the homogeneous Poisson point process (HPPP) with density λ. Following the basic property of HPPP, active and inactive users are distributed following independent HPPPs with λ^{A} and λ^{I}, respectively, where $\lambda = \lambda^{\mathrm{A}} + \lambda^{\mathrm{I}}$. Due to the symmetric property of the clustering model and HPPPs, we thereafter focus on a single cluster without loss of generality.

Users are served by the "random-push" scheduling [7], which works as follows. Considering a cluster, the BS first randomly selects an active user. Then the BS checks whether the request of the selected user can be satisfied by files in this user's own cache. If yes, the request is satisfied by self-caching. If not, the BS checks whether the request can be satisfied by files in caches of other users in the same cluster. If so, the user is served via a D2D link. Otherwise, the BS provides a BS link for service. We assume that the BS is connected to a repository that can access all required files through a unlimited backhaul. Thus, the selected user shall be served ultimately. After the scheduling of the selected user, other users check whether their requests can be satisfied by files in their own caches. If yes, their requests are satisfied. We note that although such scheduling is sub-optimal in terms of network throughput (compared to the "priority-push" in [7]), it is more mathematically tractable in many aspects, provides fairness to all users, and can serve as a reference for other more complicated systems [14].

We consider users to have different throughput when accessing a desired file using different approaches. We thus denote the throughput a user can obtain as T_{S}, T_{D}, and T_{B} when the desired file is obtained via self-caching, the D2D link, and the BS link, respectively, and consider $T_{\mathrm{S}} \geq T_{\mathrm{D}} \geq T_{\mathrm{B}}$. We note that this fixed-transmission-rate consideration is practical when equipped a fixed modulation-and-coding scheme and system-level power control. We refer the detailed arguments to Sec. II.A of [7].

We consider users to request files from a library consisting of M files. Since different users can have different preferences on files, the probability that a user k requests file m is denoted as a_m^k, where $0 \leq a_m^k \leq 1, \forall m, k$, and $\sum_{m=1}^{M} a_m^k = 1, \forall k$. We adopt the probabilistic description for the caching policy [16]. The caching policy of a user k is then described as $\{b_m^k\}_{m=1}^{M}$, where $0 \leq b_m^k \leq 1$ is the probability to cache file m and $\sum_{m=1}^{M} b_m^k \leq S$. In the remaining paper, by dropping the subscript and superscript, we will let $\{a_m^k\}$ and $\{b_m^k\}$ to be the short-hand notations for describing the individual preference and caching policy for user k, respectively.

III. Noncoordinated Individual Preference Aware Caching Policy

In the paper, we aim to design a noncoordinated individual preference aware caching policy, in which users determine their caching polices without knowing caching policies of other users and without coordination. To realize this, we propose using a group-wise caching framework. In this framework, we consider G reference groups, where each reference group $\mathcal{G}_i$ is associated with a caching policy $\{b_m^{\mathrm{Grp},i}\}$ and a group preference $\{a_m^{\mathrm{Grp},i}\}$.

When a user k appears in the D2D network, this user first determines which reference group to be associated with. Since a reference group has a caching policy, the user then

automatically adopts this caching policy. The condition for user k to be associated with reference group $\mathcal{G}_i$ is:

$$k \in \mathcal{G}_i \quad \text{if} \quad i = \arg \min_{j=1,\ldots,C} \mathcal{D}\left(\{a_m^k\} \| \{a_m^{\text{Grp},j}\}\right), \quad (1)$$

where

$$\mathcal{D}\left(\{a_m^k\} \| \{a_m^{\text{Grp},j}\}\right) = \sum_{m=1}^{M} a_m^k \log \frac{a_m^k}{a_m^{\text{Grp},j}}$$

is the KL divergence between the individual preference of user k and group preference $\{a_m^{\text{Grp},j}\}$. Since different users can obtain different caching policies using (1) according to their individual preferences and information of reference groups, users can determine their caching policies without knowing caching policies of other users and without any coordination. We note that the the information of the reference groups can be broadcast to users before the time that users start to cache files (e.g., at midnight). We will in the following discuss the design of reference groups.

IV. Learning-Aided Caching Policy Design

In this section, we first propose a learning-based reference group construction approach, and then design the caching policies associated to the groups.

A. Reference Group Construction

Assume that we have N user samples along with their individual preferences. We shall construct reference groups based on these samples, using clustering algorithms adapted from machine learning [15]. Suppose there are user samples associated to the reference group $\mathcal{G}_i$. We let the group preference distribution of $\mathcal{G}_i$ be the distribution minimizing the sum KL divergence:

$$\{a_m^{\text{Grp},i}\} = \arg \min_{\{a_m^{\text{Grp},i}\}} \sum_{k \in \mathcal{G}_i} \mathcal{D}\left(\{a_m^k\} \| \{a_m^{\text{Grp},i}\}\right), \quad (2)$$

Note that $\sum_{m=1}^{M} a_m^{\text{Grp},i} = 1$ must be satisfied by definition. Then by Karush-Kuhn-Tucker (KKT) conditions, we can prove that the group preference for reference group $\mathcal{G}_i$ is the average (mean) preferences of user samples in the group, i.e.,

$$a_m^{\text{Grp},i} = \frac{\sum_{k \in \mathcal{G}_i} a_m^k}{\sum_{n=1}^{M} \sum_{k \in \mathcal{G}_i} a_n^k}, \forall m. \quad (3)$$

To obtain the group preference, we need to determine which user sample belongs to which reference group. To do this, by using the Jensen-Shannon Divergence (JSD) of two distributions, we first define the similarity measurement for individual preferences of two different users k and l as:

$$d\left(\{a_m^k\}, \{a_m^l\}\right) = \frac{\mathcal{D}\left(\{a_m^k\} \| \{a_m^{\text{ref}}\}\right) + \mathcal{D}\left(\{a_m^l\} \| \{a_m^{\text{ref}}\}\right)}{2} \quad (4)$$

where $a_m^{\text{ref}} = \frac{1}{2}(a_m^k + a_m^l)$. Then to construct the primitive groups, the conventional agglomerative clustering [15], hierarchically grouping users in a bottom-up fashion, is used. Roughly speaking, starting with N groups, i.e., each group has only a user. The agglomerative algorithm then, at each iteration, merges two groups having the minimum group distance, where the group distance between two groups $\mathcal{G}_i$ and $\mathcal{G}_j$ is:

$$d\left(\mathcal{G}_i, \mathcal{G}_j\right) = \max_{k \in \mathcal{G}_i, l \in \mathcal{G}_j} d\left(\{a_m^k\}, \{a_m^l\}\right). \quad (5)$$

Thus, to construct G groups, $N - G$ iterations would be taken. Since agglomerative clustering is commonly used in machine learning for unsupervised learning, we omit the details and refer to Ch. 10 of [15] for brevity.

After obtaining the grouping result (with G groups) of agglomerative clustering, we further refine the grouping by using a K-means clustering [15], in which the aim is to minimize the sum KL divergence between user samples and their corresponding group preference distributions. Notice that, on one hand, when given a clustering of users, results in (2) and (3) indicate that the mean point of the individual preferences of users in a group minimizes the sum KL divergences. On the other hand, when given the group preferences, the group assignment in (1) minimizes the KL divergences between user samples and their associated group preference distributions. By using these two properties, we propose a K-means clustering that iteratively conducts the computations of group preferences using (2) and re-assigns the user samples to different groups by (1). Since each iteration improves the sum KL divergences of users, this iterative algorithm can converge to the local optimum minimizing the sum KL divergence of the sample users. The final grouping result is then the input for designing the caching policy of each group, which is discussed in the subsequent subsection. We again refer the details of the K-means clustering to Ch. 10 of [15].

B. Throughput and Hit-Rate Expressions

After the construction of reference groups. We then design their corresponding caching policies that optimize particular objective functions. As a first step we here derive the objective functions for throughput and hit-rate optimizations.

We first derive some fundamental access probabilities that will be used later in the derivations. Suppose the number of users assigned to the reference group $\mathcal{G}_i$ in a cluster is $n_i, i = 1, 2, ..., G$. Considering a user assigned to group $\mathcal{G}_i$ with the individual preference *approximated* by the group preference, the probability that the requests of such user can be satisfied by self-caching is: $P_{\text{S}}^{\text{Grp},i} = \sum_{m=1}^{M} a_m^{\text{Grp},i} b_m^{\text{Grp},i}$. The probability that the user cannot find the desired files from caches of users in the cluster and thus has to resort to a BS link is: $P_{\text{B}}^{\text{Grp},i} = \sum_{m=1}^{M} a_m^{\text{Grp},i} \prod_{l=1}^{G} \left(1 - b_m^{\text{Grp},l}\right)^{n_l}$. By using above results, the probability that the user can find the desired files through a D2D link is: $P_{\text{D}}^{\text{Grp},i} = 1 - P_{\text{B}}^{\text{Grp},i} - P_{\text{S}}^{\text{Grp},i}$.

We start to derive the network throughput. We first estimate the densities of users assigned to each group by:

$$\lambda_i^{\text{A}} = \frac{\lambda^{\text{A}} |\mathcal{G}_i|}{\sum_{j=1}^{G} |\mathcal{G}_j|} \quad ; \quad \lambda_i^{\text{I}} = \frac{\lambda^{\text{I}} |\mathcal{G}_i|}{\sum_{j=1}^{G} |\mathcal{G}_j|}, \quad (6)$$

where λ_i^{I} is the density of active users for group $\mathcal{G}_i$, λ_i^{I} is the density of inactive users for group $\mathcal{G}_i$, and $|\mathcal{G}_i|$ is the number

of user samples belongs to reference group $\mathcal{G}_i$. Then due to HPPP model, the mean numbers of active and inactive users in a cluster are $\kappa_i^{\mathrm{A}} = \lambda_i^{\mathrm{A}} D^2$ and $\kappa_i^{\mathrm{I}} = \lambda_i^{\mathrm{I}} D^2$, respectively. We also let $\kappa_i = \kappa_i^{\mathrm{A}} + \kappa_i^{\mathrm{I}}, \forall i$, $\kappa^{\mathrm{A}} = \sum_{i=1}^{G} \kappa_i^{\mathrm{A}}$, $\kappa^{\mathrm{I}} = \sum_{i=1}^{G} \kappa_i^{\mathrm{I}}$, and $\kappa = \kappa^{\mathrm{A}} + \kappa^{\mathrm{I}}$. Since we consider the random-push network, the expected throughput of the network is

$$T_{\text{net}} = T_{\text{sele}} + T_{\text{self}}, \tag{7}$$

where T_{sele} is the expected throughput of the selected user, and T_{self} is the expected throughput from the self-caching of other users. Denote n_i^{A} as the number of active users and n_i^{I} as the number of inactive users for group $\mathcal{G}_i$ in a cluster. Then

$$T_{\text{sele}} = \mathbb{E}\left[\frac{\sum_{g=1}^{G} n_g^{\mathrm{A}} T_g}{\sum_{g=1}^{G} n_g^{\mathrm{A}}}\right] = \Pr(n_1^{\mathrm{A}} + ... + n_G^{\mathrm{A}} > 0) \cdot \underbrace{\sum_{\substack{n_1^{\mathrm{A}}, n_1^{\mathrm{I}}, ..., n_G^{\mathrm{I}} \\ n_1^{\mathrm{A}} + ... + n_G^{\mathrm{A}} > 0}} \left[\frac{\sum_{g=1}^{G} n_g^{\mathrm{A}} T_g}{\sum_{g=1}^{G} n_g^{\mathrm{A}}} \frac{\Pr(n_1^{\mathrm{A}}, n_1^{\mathrm{I}}, ..., n_G^{\mathrm{I}})}{\Pr(n_1^{\mathrm{A}} + ... + n_G^{\mathrm{A}} > 0)}\right]}_{(i)}. \tag{8}$$

Since we consider HPPP, we obtain $\Pr(n_1^{\mathrm{A}} +, ..., + n_G^{\mathrm{A}} > 0) = 1 - e^{-\kappa^{\mathrm{A}}}$. To compute (i) of (8), the approximation that $\mathbb{E}\left[\frac{x}{y}\right] \approx \frac{\mathbb{E}[x]}{\mathbb{E}[y]}$ is used. We then obtain the result in (9) on the top of next page, where (a) is due to $T_g = 0$ when $n_1^{\mathrm{A}} = n_2^{\mathrm{A}} = ... = n_G^{\mathrm{A}} = 0$ and

$$T_g = T_{\mathrm{B}} P_{\mathrm{B}}^{\text{Grp},i} + T_{\mathrm{D}} P_{\mathrm{D}}^{\text{Grp},i} + T_{\mathrm{S}} P_{\mathrm{S}}^{\text{Grp},i}. \tag{10}$$

We then turn to compute T_{self}. Since T_{self} comes from those users that are not selected, by using $P_{\mathrm{S}}^{\text{Grp},i}$ and the similar approximation in (9), we obtain the result in (11) on the top of next page. Finally, by combining (7), (9), and (11), and after some algebraic manipulations, we obtain the approximated T_{net} in (12) on the top of next page, where $L_i^{\mathrm{A}} = \frac{\kappa_i^{\mathrm{A}}}{\kappa^{\mathrm{A}}}$.

The hit-rate of the network is defined as the probability that a user can find the desired files in a cluster without using a BS link. Thus, the hit-rate of the network is

$$H_{\text{net}} = \mathbb{E}\left[\frac{\sum_{i=1}^{G} n_i^{\mathrm{A}} \left(1 - P_{\mathrm{B}}^{\text{Grp},i}\right)}{\sum_{i=1}^{G} n_i^{\mathrm{A}}}\right]. \tag{13}$$

By using the similar approach, we can obtain the approximation of the hit-rate of the network:

$$H_{\text{net}} \approx (1 - e^{-\kappa^{\mathrm{A}}}) \cdot \left(1 - \sum_{m=1}^{M} \sum_{i=1}^{G} L_i^{\mathrm{A}} a_m^{\text{Grp},i} (1 - b_m^{\text{Grp},i}) \prod_{j=1}^{G} e^{-\kappa_j b_m^{\text{Grp},j}}\right). \tag{14}$$

C. Caching Policy Design

Here we propose an algorithm that can be used to maximize T_{net} and H_{net}. In the following, we consider maximizing T_{net} as an example, and the approach can be similarly applied to maximizing H_{net}. We observe that directly maximizing T_{net} by jointly optimizing caching policies of all groups is difficult, due to the complicated production term in (12). In contrast, when optimizing the caching policy of group $\mathcal{G}_i$ while fixing policies of the other groups, the maximization problem can be simplified as

$$\max_{b_m^{\text{Grp},i}, \forall m} T_i^{\text{sub}} \quad \text{s.t.} \sum_{m=1}^{M} b_m^{\text{Grp},i} \leq S, 0 \leq b_m^{\text{Grp},i} \leq 1, \forall m, \tag{15}$$

where T_i^{sub} is provided in (16) on the top of next page. Since the Hessian of T_i^{sub} is negative semi-definite, (15) is a concave optimization problem. Then by solving the sub-problems iteratively for different groups until convergence, the caching policies of different groups are obtained. Note that the proposed iterative algorithm falls in the framework of the block coordinate descent (BCD) approach; the convergence analysis of the BCD approach can be found in [14].

V. Numerical Results

Numerical results are provided in this section to evaluate the performance of the proposed design. Because it is more practical that users do not have requests all the time, we consider active and inactive users distributed with densities $\lambda^{\mathrm{A}} = 0.002\ \mathrm{m}^{-2}$ and $\lambda^{\mathrm{I}} = 0.008\ \mathrm{m}^{-2}$, respectively. We consider 20 MHz of bandwidth for each D2D link, which is practically realizable when adopting mmWave systems with reuse factor 16 or conventional systems with reuse factor one using advanced MIMO approaches to mitigate inter-cluster interference. A BS link with 20 kHz of bandwidth per user always exists when it is needed. We assume that the 2.06 bits/s/Hz spectral efficiency, corresponding to 5 dB of signal-to-noise ratio, is guaranteed for both D2D and BS links. Thus, the throughput for a D2D link is $T_{\mathrm{D}} = 41.2$ Mbits/s; for a BS link is $T_{\mathrm{B}} = 41.2$ kbits/s. We then consider $T_{\mathrm{S}} = 2T_{\mathrm{D}}$, indicating the slightly better video quality a user can have if the file is obtained directly from the local cache (note that while the "transmission rate" from the cache to the user is very high for a self-cached file, it is the playback rate that determines the effective throughput).

We consider $S = 5$ for all users, and $G = 32$ is used by the proposed design. To evaluate with practical individual preference probabilities, we use an individual preference generator parameterized by two different datasets: (1) GlobeCom dataset in [9] with $M = 467$; and (2) ToN-June dataset in [10] with $M = 500$. The main difference between them is that the GlobeCom dataset only considers users with higher traffic load. In all evaluations, for each dataset, we first generate 20000 user samples. Among them, 3000 samples are used for designing the reference groups and their caching policies. All 20000 samples are later used for obtaining simulation results. The global popularity distribution of the users in those evaluations is thus by definition the average of the individual preferences of 20000 users.

The derived approximations in (12) and (14) are validated by comparisons with simulation results in Fig. 1. Different from the other figures, to focus on validating the expressions, here we let user preferences to be identical to their associated group preference when obtaining the simulating curves. This

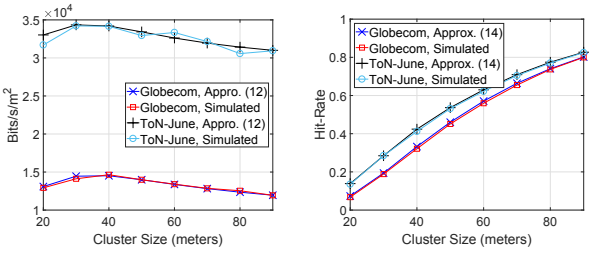
$$\sum_{\substack{n_1^{\mathrm{A}},n_1^{\mathrm{I}},\ldots,n_G^{\mathrm{I}}\\ n_1^{\mathrm{A}}+\ldots+n_G^{\mathrm{A}}>0}} \left[\frac{\sum_{g=1}^{G} n_g^{\mathrm{A}} T_g}{\sum_{g=1}^{G} n_g^{\mathrm{A}}} \frac{\Pr(n_1^{\mathrm{A}}, n_1^{\mathrm{I}}, ..., n_G^{\mathrm{I}})}{\Pr(n_1^{\mathrm{A}}+...+n_G^{\mathrm{A}}>0)}\right] \approx \left[\frac{\sum\limits_{\substack{n_1^{\mathrm{A}},n_1^{\mathrm{I}},\ldots,n_G^{\mathrm{I}}\\ n_1^{\mathrm{A}}+\ldots+n_G^{\mathrm{A}}>0}} \sum\limits_{g=1}^{G} n_g^{\mathrm{A}} T_g \frac{\Pr(n_1^{\mathrm{A}}, n_1^{\mathrm{I}}, ..., n_G^{\mathrm{I}})}{1-e^{-\kappa^{\mathrm{A}}}}}{\sum\limits_{\substack{n_1^{\mathrm{A}},n_1^{\mathrm{I}},\ldots,n_G^{\mathrm{I}}\\ n_1^{\mathrm{A}}+\ldots+n_G^{\mathrm{A}}>0}} \sum\limits_{g=1}^{G} n_g^{\mathrm{A}} \frac{\Pr(n_1^{\mathrm{A}}, n_1^{\mathrm{I}}, ..., n_G^{\mathrm{I}})}{1-e^{-\kappa^{\mathrm{A}}}}}\right] \tag{9}$$

$$\overset{(a)}{=} \left[\frac{\sum\limits_{n_1^{\mathrm{A}},n_1^{\mathrm{I}},\ldots,n_G^{\mathrm{I}}} \sum\limits_{g=1}^{G} n_g^{\mathrm{A}} T_g \Pr(n_1^{\mathrm{A}}, n_1^{\mathrm{I}}, ..., n_G^{\mathrm{I}})}{\sum\limits_{n_1^{\mathrm{A}},n_1^{\mathrm{I}},\ldots,n_G^{\mathrm{I}}} \sum\limits_{g=1}^{G} n_g^{\mathrm{A}} \Pr(n_1^{\mathrm{A}}, n_1^{\mathrm{I}}, ..., n_G^{\mathrm{I}})}\right] = \frac{\mathbb{E}\left[\sum_{g=1}^{G} n_g^{\mathrm{A}} T_g\right]}{\mathbb{E}\left[\sum_{g=1}^{G} n_g^{\mathrm{A}}\right]} = \frac{\sum_{g=1}^{G} \mathbb{E}\left[n_g^{\mathrm{A}} T_g\right]}{\sum_{g=1}^{G} \kappa_g^{\mathrm{A}}}.$$

$$T_{\mathrm{Self}} = T_{\mathrm{S}} \mathbb{E}\left[\sum_{m=1}^{M}\sum_{i=1}^{G} a_m^{\mathrm{Grp},i} b_m^{\mathrm{Grp},i} \left(n_i - \mathbf{1}_{\{i,\mathrm{sel}\}}\right)\right] \approx T_{\mathrm{S}} \sum_{m=1}^{M}\sum_{i=1}^{G} \kappa_i^{\mathrm{A}} a_m^{\mathrm{Grp},i} b_m^{\mathrm{Grp},i} - T_{\mathrm{S}} \frac{\sum_{m=1}^{M}\sum_{i=1}^{G} \kappa_i^{\mathrm{A}} a_m^{\mathrm{Grp},i} b_m^{\mathrm{Grp},i}}{\sum_{i=1}^{G} \kappa_i^{\mathrm{A}}} (1-e^{-\kappa^{\mathrm{A}}}). \tag{11}$$

$$\begin{aligned} T_{\mathrm{net}} \approx & (1-e^{-\kappa^{\mathrm{A}}}) \left(T_{\mathrm{D}} + (T_{\mathrm{B}} - T_{\mathrm{D}}) \sum_{m=1}^{M}\sum_{i=1}^{G} L_i^{\mathrm{A}} a_m^{\mathrm{Grp},i} (1-b_m^{\mathrm{Grp},i}) \prod_{j=1}^{G} e^{-\kappa_j b_m^{\mathrm{Grp},j}}\right) \\ & + \sum_{m=1}^{M}\sum_{i=1}^{G} (T_{\mathrm{S}}\kappa_i^{\mathrm{A}} - L_i^{\mathrm{A}} T_{\mathrm{D}} + e^{-\kappa^{\mathrm{A}}} L_i^{\mathrm{A}} T_{\mathrm{D}}) a_m^{\mathrm{Grp},i} b_m^{\mathrm{Grp},i} \end{aligned} \tag{12}$$

$$\begin{aligned} & T_i^{\mathrm{sub}} = (1-e^{-\kappa^{\mathrm{A}}})(T_{\mathrm{B}} - T_{\mathrm{D}}) \sum_{m=1}^{M} \left[C_m^{i,1} + C_m^{i,2}(1-b_m^{\mathrm{Grp},i})\right] e^{-\kappa_i b_m^{\mathrm{Grp},i}} + (T_{\mathrm{S}}\kappa_i^{\mathrm{A}} - L_i^{\mathrm{A}} T_{\mathrm{D}} + e^{-\kappa^{\mathrm{A}}} L_i^{\mathrm{A}} T_{\mathrm{D}}) \sum_{m=1}^{M} a_m^{\mathrm{Grp},i} b_m^{\mathrm{Grp},i}; \\ & C_m^{i,1} = \left(\sum_{j=1,j\neq i}^{G} L_j^{\mathrm{A}} a_m^{\mathrm{Grp},j} (1-b_m^{\mathrm{Grp},j})\right) \left(\prod_{j\neq i} e^{-\kappa_j b_m^{\mathrm{Grp},j}}\right); C_m^{i,2} = L_i^{\mathrm{A}} a_m^{\mathrm{Grp},i} \left(\prod_{j\neq i} e^{-\kappa_j b_m^{\mathrm{Grp},j}}\right). \end{aligned} \tag{16}$$



(a) Throughput. (b) Hit-Rate.

Fig. 1: Comparisons between analyses and simulations.

avoids the error caused by the deviations between the user preferences and the group preferences. The curves in Fig. 1(a) are generated using the proposed distributed design optimizing throughput; the curves in Fig. 1(b) optimize hit-rate. We observe that the simulation results are close to the analytical results, which validates our approximations.

In the remaining figures, we evaluate the proposed designs optimizing throughput and hit-rate, labeled with "Nco. Throughput" and "Nco. Hit-Rate", respectively. We compare them with other reference designs: (i) coordinated individual preference aware designs, aiming to optimize throughput or hit-rate, proposed in [14] (labeled with "Co. Throughput/Hit-Rate"); (ii) the "Selfish" policy in which each user caches their most desired files without considering other users; and (iii) the designs proposed in [7], aiming to optimize throughput or hit-rate based merely on the global popularity distribution (labeled with "Glo. Throughput/Hit-Rate"). Considering the GlobeCom dataset, the performance evaluation in terms of area throughput is provided in Fig. 2. We observe that the proposed design with throughput optimization is better than the selfish policy and designs using a global popularity distribution. In Fig. 3, the same evaluation is provided for hit-rate. The results again show that the proposed design outperforms the selfish policy significantly and is slightly better than designs using a global popularity distribution. This indicates that to have a good hit-rate performance, knowing the average probability for a file to be requested in the network is more important than specific which files are more preferable to whom. From Figs. 2 and 3, we observe that the distributed designs are worse than the coordinated designs as expected. However, the benefit of a distributed design is the ability to implement without coordination between users.

In Figs. 4 and 5, we conduct the same evaluations as in Figs. 2 and 3 but with ToN-June dataset. The results are slightly different from the GlobeCom dataset in that the selfish policy this time is better than the proposed design. The reasons are: (i) the hit-rate performance of selfish design in ToN-June dataset

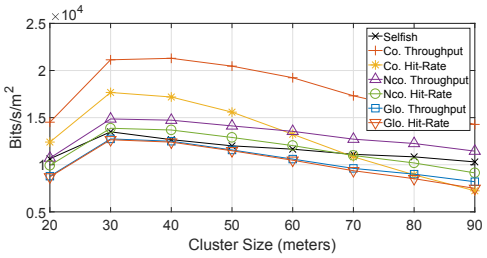


Fig. 2: Throughput evaluation for D2D caching networks with GlobeCom dataset.

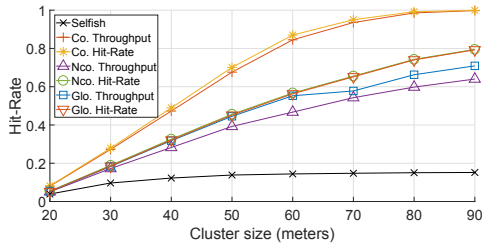


Fig. 3: Hit-Rate evaluation for D2D caching networks with GlobeCom dataset.

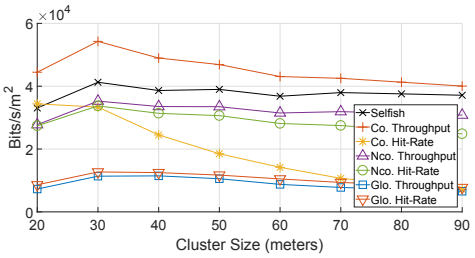


Fig. 4: Throughput evaluation for D2D caching networks with ToN-June dataset.

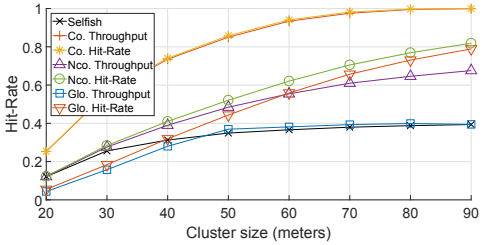


Fig. 5: Hit-Rate evaluation for D2D caching networks with ToN-June dataset.

is inherently good; and (ii) the selfish policy actually has a higher heterogeneity than the proposed distributed design, as each user can have their own caching policies when the selfish policy is adopted. Such result motivates us to investigate how we can improve the heterogeneity of the proposed design by combining the concept of the selfish policy with our distributed design, as one of our future directions. In addition to the above, other observations are similar to Figs. 2 and 3, i.e., the proposed design outperforms designs using the global popularity distribution and is worse than the coordinated designs. Overall, the simulation results indicate that our distributed design can benefit from the information of individual preference without the need for centralized cooperation of users. Besides, although not shown here, the performance of the distributed design generally degrades when decreasing the number of groups G, and ultimately becomes almost identical to designs based on the global popularity distribution.

VI. Conclusions

In this paper, we proposed a noncoordinated individual preference aware caching policy for cache-aided D2D networks. This caching policy is based on a group-wise structure that allows users, according to their own preferences, to be assigned to a reference group with the uniquely associated caching policy without any coordination. Machine learning approaches were exploited to construct reference groups, and problems for optimizing network throughput and hit-rate were formulated and solved. Numerical results based on practical individual preferences showed that the proposed distributed caching policy can benefit from the information of individual preference without the need of coordination of users.

References

[1] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 16-22, Aug. 2016.

[2] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Device-to-device collaboration through distributed storage," *in Proc. IEEE GLOBECOM*, Dec. 2012.

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

[4] M.-C. Lee, M. Ji, A. F. Molisch, and N. Sastry, "Throughput-Outage analysis and evaluation of cache-aided D2D networks with measured popularity distributions," *IEEE Trans. Wireless Commun.,* vol. 18, no. 11, pp. 5316-5332, Nov. 2019.

[5] T. Deng, G. Ahani, P. Fan, 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.

[6] N. Deng and M. Haenggi, "The benefits of hybrid caching in Gauss-Poisson D2D networks," *IEEE Trans. Wireless Commun.,* vol. 17, no. 1, pp. 492-505, Jan. 2018.

[7] M.-C. Lee and A. F. Molisch, "Caching policy and cooperation distance design for base station assisted wireless D2D caching networks: Throughput and energy efficiency optimization and trade-off," *IEEE Trans. Wireless Commun.,* vol. 17, no. 11, pp. 7500-7514, Nov. 2018.

[8] L. Shi, L. Zhao, G. Zheng, Z. Han, Y. Ye, "Incentive design for cache-enabled D2D underlaid cellular networks using Stackelberg game," *IEEE Trans. Veh. Technol.,* vol. 68. no. 1, pp. 765-779, Jan. 2019.

[9] M.-C. Lee, A. F. Molisch, N. Sastry, and A. Raman, "Individual preference probability modeling for video content in wireless caching networks," *in Proc. IEEE GLOBECOM*, Dec. 2017.

[10] M.-C. Lee, A. F. Molisch, N. Sastry, and A. Raman, "Individual preference probability modeling and parameterization for video content in wireless caching networks," *IEEE/ACM Trans. Netw.,*, Mar., 2019.

[11] 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.

[12] B. Chen and C. Yang, "Caching policy optimization for D2D communications by learning user preference," *in Proc. IEEE VTC-Spring*, Jun. 2017.

[13] Y. Li, C. Zhong, M. C. Gursory, and S. Velipasalar, "Learning-Based delay-aware caching in Wireless D2D caching networks," *IEEE Access,* vol. 6, pp. 77250-77264, Nov. 2018.

[14] M.-C. Lee and A. F. Molisch, "Individual preference aware caching policy design in wireless D2D networks," *submitted*. Online Available: arXiv:1903.02705, Mar. 2019.

[15] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification,* John Wiley & Sons Inc., 2nd ed., 2006.

[16] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," *in Proc. IEEE ICC*, Jun. 2015.