

Coded Caching in Partially Cooperative D2D Communication Networks

Ali Tebbi, Chi Wan Sung
Department of Electronic Engineering
City University of Hong Kong
Email: ali.tebbi, albert.sung@cityu.edu.hk

Abstract—The backhaul traffic is becoming a major concern in wireless and cellular networks (e.g., 4G-LTE and 5G) with the increasing demands for online video streaming. Caching the popular content in the cache memory of the network users (e.g., mobile devices) is an effective technique to reduce the traffic during the networks’ peak time. However, due to the dynamic nature of these networks, users privacy settings, or energy limitations, some users may not be available or intend to participate during the caching procedures. In this paper, we propose caching schemes for device-to-device communication networks where a group of users show selfish characteristics. The selfish users along with the non-selfish users will cache the popular content, but will not share their useful cache content with the other users to satisfy a user request. We show that our proposed schemes are able to satisfy any arbitrary user requests under partial cooperation of the network users.

Index Terms—Cache networks, D2D networks, Selfish user

I. INTRODUCTION

The data traffic in wireless and mobile networks is exponentially growing during the recent years specially due to the expanding demands of multimedia contents such as YouTube videos [1]. The massive resultant backhaul traffic in wireless networks can be reduced by bringing the popular content forward closer to the edge of the network. This technique is called “caching” which basically is storing the popular content in resources closer to the end users such as base stations, routers, and mobile devices in order to serve the users’ requests. A caching scheme generally consists of two phases: caching (or placement) phase and delivery phase. The caching phase carries out during the off-peak time of the network when the network is not congested to duplicate and distribute the popular content across the network. The delivery phase is carried out during the peak time of the network when the network is congested in order to serve the users’ requests by the contents cached in the network.

A novel approach of caching is recently introduced by Maddah-Ali *et al.* in [2] known as “coded caching” where it aims to jointly optimize the caching and delivery phase in order to achieve a global caching gain (i.e., caching gain which is attained from aggregate global cache size). The cache placement in this approach is based on constructing a multicast opportunity for all users at the same time. In other words, the

server can deliver any arbitrary users demands by a number of coded multicast transmissions.

Since the introduction of the coded caching, this approach received a considerable attention due to its noticeable caching gain (i.e., achieving local and global gain at the same time) over the conventional schemes. The centralized nature of the caching scheme in [2] is quite far from the real wireless communication networks. For example, due to the mobility in cellular networks, the number of active users during the delivery phase might not be the same as in the caching phase. A “decentralized” approach of coded caching is proposed in [3] where during the placement phase each user randomly and independently caches some parts of each file. Then, during the delivery phase, the server (or base station) takes the advantage of the multicast opportunities which has been created during the caching phase to satisfy the user requests.

Several works can be found in literature studying caching schemes for Device-to-Device (D2D) communication networks [4]–[8]. A one-hop D2D network is considered in [5] where each user caches a subset of files randomly. It has been shown that the throughput per user varies as a function of the user cache size over the file library size under this scheme for sufficiently large number of users and files. The coded caching for D2D communication networks is studied in [7]. In the proposed model, there exists no central server (or base station) in the network to control the caching and delivery phases while all users contribute by multicasting their useful cache content to other users during the delivery phase in order to satisfy any user request.

In cellular networks it is likely that some of the users have left the network during the delivery phase which makes their cache content not available for the other users during the delivery phase. Moreover, it is likely that some of the users will not participate during the delivery phase due to their privacy settings or their energy saving concerns.

In this paper, we consider a partially cooperative D2D communication network without a central server (or base station) where some of the users are selfish. More precisely, although all of the users will cache the file library in their cache memory, only a group of non-selfish users participate in delivery phase by transmitting their useful cache content. Obviously, all of the users including selfish users can receive these transmissions to recover their requested file. We propose a “deterministic” and “random” caching scheme where we

This work was partially supported by a grant from the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08).

show that these schemes can satisfy any arbitrary user requests at the presence of selfish users in the network (i.e., partial cooperation of the users). We also establish the necessary conditions and achievable rates of these schemes.

The remainder of this paper is organised as follows. In Section II, we introduce the network model and the mathematical tools and definitions which we will be using in our proposed caching schemes. In Sections III and IV we present our deterministic and random caching schemes for D2D networks with selfish users, respectively. We discuss the caching and delivery procedures and provide examples, the corresponding necessary conditions and achievable rates. We then conclude the paper in Section V.

II. PROBLEM STATEMENT

Our proposed network consists of K users $\mathcal{U} = \{u_1, \dots, u_K\}$ each with a cache memory of size MB bits. The users of the network are divided into two groups of *selfish* and *non-selfish* users specified by the subsets \mathcal{S} and \mathcal{T} , respectively, such that $\mathcal{U} = \mathcal{S} \cup \mathcal{T}$. All users are in the coverage range of each other such that they all can communicate with each other.

Definition 1 (File Library). *The file library is a set of N popular demanding files (e.g. videos) $\Omega = \{\omega_1, \dots, \omega_N\}$ each of size B bits.*

We assume that each user u_i makes an arbitrary request r_k from the file library Ω independent from the other users. Denote the request vector of all users by $\mathcal{R} = (r_1, r_2, \dots, r_K)$, where $r_k \in \Omega$ for all $k \in \{1, \dots, K\}$.

Definition 2 (Caching Function). *The caching function consists of a function ϕ_u to map the file library into the cache memory of user u such that $\phi_u : \mathbb{F}_2^{NB} \rightarrow \mathbb{F}_2^{MB}$.*

Definition 3 (Delivery Functions). *The delivery functions consist of two functions $\psi_u : \mathbb{F}_2^{MB} \times \Omega^K \rightarrow \mathbb{F}_2^{R_u B}$ and $\Psi_u : \mathbb{F}_2^{B \sum_{v \in \mathcal{T}_u} R_v} \times \mathbb{F}_2^{MB} \times \Omega^K \rightarrow \mathbb{F}_2^B$, respectively, for generating the transmitted message of user u and for decoding the received messages at user u , where \mathcal{T}_u is the set of users whose messages are received at user u and R_u is the transmission rate of user u .*

The caching and delivery functions are the essential tools to characterize the caching and delivery procedures as follows.

Caching Phase: During the caching phase, the cache content Γ_u of each user u is generated and stored in its cache memory by employing the caching function such that $\Gamma_u = \phi_u(\omega_n : \forall \omega_n \in \Omega)$.

Delivery Phase: During the delivery phase, each non-selfish user u generates its transmitted message $X_{u, \mathcal{R}} = \psi_u(\Gamma_u, \mathcal{R})$ as a function of its cache content Γ_u and the user request vector \mathcal{R} . Furthermore, each user u decodes its requested file $\hat{\omega}_{u, \mathcal{R}} = \Psi_u(\{X_{v, \mathcal{R}} : \forall v \in \mathcal{T}_u\}, \Gamma_u, \mathcal{R})$ as a function of the received messages $X_{v, \mathcal{R}}$ from all the transmitter nodes $v \in \mathcal{T}_u$, its cache content Γ_u , and the request vector \mathcal{R} .

During the caching phase all of the users in \mathcal{U} will generate their cache content as a function of the file library Ω . However, only non-selfish users in \mathcal{T} will participate during the delivery phase. Then, the worst-case error probability is given by

$$\Pr_e = \max_{\mathcal{R}} \max_{u \in \mathcal{U}} \mathbb{P}(\hat{\omega}_{u, \mathcal{R}} \neq \omega_u).$$

Definition 4 (Achievable rate). *Let $R = \sum_{u \in \mathcal{U}} R_u$ where R_u is the transmission rate of non-selfish user u to satisfy a request vector \mathcal{R} . Then, the Memory-Rate pair (M, R) (i.e., $R(M)$) is achievable if $R(M)$ is achievable for any arbitrary request vector \mathcal{R} . In other words, the rate $R(M)$ is said to be achievable if for any $\epsilon > 0$ and sufficiently large file size B there exists a caching-delivery scheme such that $\Pr_e \leq \epsilon$.*

III. DETERMINISTIC CACHING

In this section we propose a deterministic caching scheme where each user will cache a predetermined parts of each file. As it is proposed in [2] and [7], deterministic caching procedure will create a multicasting opportunity among all users during the delivery phase in order to retrieve their requested file. As mentioned earlier, only non-selfish users will transmit their cache content during the delivery phase. Let $t = \frac{MK}{N}$ be an integer between 1 and $K - 1$. Note that when $t \geq K$, each user can cache all the N files.

• **Caching phase:** The caching procedure consists of the following steps:

- 1) Each file $\omega_n \in \Omega$ is divided into $t \binom{K}{t}$ subfiles.
- 2) The subfiles of each file are partitioned into $\binom{K}{t}$ groups, each of which has t subfiles. Each group is indexed by a specific subset of t users.
- 3) For every file in the library, its subfiles in each group will be cached in the corresponding subset of users. For example, the subfiles in the group with index $\{u_{i_1}, u_{i_2}, \dots, u_{i_t}\}$ will be cached in users $u_{i_1}, u_{i_2}, \dots, u_{i_t}$.

This caching procedure guarantees that any subset of t users share the same t subfiles of each file in the library. Furthermore, each user caches a total of $\binom{K-1}{t-1} t N$ subfiles. Since each subfile is of size $B / \binom{K}{t}$ bits, the total number of bits stored in the cache of each user is $tNB/K = MB$, which satisfies the cache memory constraint.

Example 1. *Consider a network of $K = 4$ users each of which has cache size $M = 3$ and a file library $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ of size $N = 4$ files. Since $t = 3$, then each file are divided into 12 subfiles and any subset of $t = 3$ users will cache the same 3 subfiles from each file. The cache content of each user is depicted in Figure 1 which follows from the aforementioned caching phase.*

In contrast to the delivery phases proposed in [3] and [7] where a base station has access to the whole file library and is responsible for all the transmission or a fully cooperative users during the delivery phase, in our proposed scheme, a group of selfish users will not participate during the delivery phase in order to transmit their useful cache contents. Therefore, the

u_1	u_2
$\omega_{1,1}, \omega_{1,2}, \omega_{1,3}, \omega_{1,4}, \omega_{1,5}, \omega_{1,6}, \omega_{1,7}, \omega_{1,8}, \omega_{1,9}$	$\omega_{1,1}, \omega_{1,2}, \omega_{1,3}, \omega_{1,4}, \omega_{1,5}, \omega_{1,6}, \omega_{1,10}, \omega_{1,11}, \omega_{1,12}$
$\omega_{2,1}, \omega_{2,2}, \omega_{2,3}, \omega_{2,4}, \omega_{2,5}, \omega_{2,6}, \omega_{2,7}, \omega_{2,8}, \omega_{2,9}$	$\omega_{2,1}, \omega_{2,2}, \omega_{2,3}, \omega_{2,4}, \omega_{2,5}, \omega_{2,6}, \omega_{2,10}, \omega_{2,11}, \omega_{2,12}$
$\omega_{3,1}, \omega_{3,2}, \omega_{3,3}, \omega_{3,4}, \omega_{3,5}, \omega_{3,6}, \omega_{3,7}, \omega_{3,8}, \omega_{3,9}$	$\omega_{3,1}, \omega_{3,2}, \omega_{3,3}, \omega_{3,4}, \omega_{3,5}, \omega_{3,6}, \omega_{3,10}, \omega_{3,11}, \omega_{3,12}$
$\omega_{4,1}, \omega_{4,2}, \omega_{4,3}, \omega_{4,4}, \omega_{4,5}, \omega_{4,6}, \omega_{4,7}, \omega_{4,8}, \omega_{4,9}$	$\omega_{4,1}, \omega_{4,2}, \omega_{4,3}, \omega_{4,4}, \omega_{4,5}, \omega_{4,6}, \omega_{4,10}, \omega_{4,11}, \omega_{4,12}$

u_3	u_4
$\omega_{1,1}, \omega_{1,2}, \omega_{1,3}, \omega_{1,7}, \omega_{1,8}, \omega_{1,9}, \omega_{1,10}, \omega_{1,11}, \omega_{1,12}$	$\omega_{1,4}, \omega_{1,5}, \omega_{1,6}, \omega_{1,7}, \omega_{1,8}, \omega_{1,9}, \omega_{1,10}, \omega_{1,11}, \omega_{1,12}$
$\omega_{2,1}, \omega_{2,2}, \omega_{2,3}, \omega_{2,7}, \omega_{2,8}, \omega_{2,9}, \omega_{2,10}, \omega_{2,11}, \omega_{2,12}$	$\omega_{2,4}, \omega_{2,5}, \omega_{2,6}, \omega_{2,7}, \omega_{2,8}, \omega_{2,9}, \omega_{2,10}, \omega_{2,11}, \omega_{2,12}$
$\omega_{3,1}, \omega_{3,2}, \omega_{3,3}, \omega_{3,7}, \omega_{3,8}, \omega_{3,9}, \omega_{3,10}, \omega_{3,11}, \omega_{3,12}$	$\omega_{3,4}, \omega_{3,5}, \omega_{3,6}, \omega_{3,7}, \omega_{3,8}, \omega_{3,9}, \omega_{3,10}, \omega_{3,11}, \omega_{3,12}$
$\omega_{4,1}, \omega_{4,2}, \omega_{4,3}, \omega_{4,7}, \omega_{4,8}, \omega_{4,9}, \omega_{4,10}, \omega_{4,11}, \omega_{4,12}$	$\omega_{4,4}, \omega_{4,5}, \omega_{4,6}, \omega_{4,7}, \omega_{4,8}, \omega_{4,9}, \omega_{4,10}, \omega_{4,11}, \omega_{4,12}$

Fig. 1: The users cache memory content of a network with $K = 4$ users of cache size $M = 3$ and a file library of size $N = 4$ with deterministic caching.

non-selfish users have to compensate for the selfishness of the selfish nodes through transmitting more useful information to other users. Our scheme can tolerate at most $t-1$ selfish users in the network.

• **Delivery phase:** Let $\mathbf{U} \subset \mathcal{U}$ be a subset of $t+1$ users. Furthermore, let $\mathbf{U} = \mathcal{S} \cup \mathcal{T}$, where \mathcal{S} and \mathcal{T} are the sets of selfish and non-selfish users in \mathbf{U} , and $|\mathcal{S}| \leq t-1$. Consider a user $u_k \in \mathbf{U}$. The subset of users $\mathbf{U} \setminus \{u_k\}$ shares t subfiles that are useful for u_k in order to retrieve its requested file. Denote the set of these t subfiles by $\mathcal{P}_{\mathbf{U},k}$. If user u_k obtained all subfiles in $\mathcal{P}_{\mathbf{U},k}$, for all $\mathbf{U} \subset \mathcal{U}$ with $u_k \in \mathbf{U}$ and $|\mathbf{U}| = t+1$, then it would have a total of $\binom{K-1}{t}t$ useful subfiles. Since there are $\binom{K-1}{t-1}t$ subfiles of its requested file in its own cache memory, user u_k would have obtained its whole requested file. In other words, user u_k needs to obtain one distinct subfile from each user in $\mathbf{U} \setminus \{u_k\}$. This implies that each user $u_i \in \mathbf{U}$ can provide one useful subfile to each user in $\mathbf{U} \setminus \{u_i\}$, and these t useful subfiles that user u_i can deliver are distinct.

We consider the following cases for all $\mathbf{U} \subset \mathcal{U}$ of size $|\mathbf{U}| = t+1$:

- 1) First, $\mathcal{S} = \emptyset$. Then, $\mathcal{T} = \mathbf{U}$. In this case, to deliver the t useful subfiles to each user in \mathbf{U} , each user $u_i \in \mathcal{T}$ will multicast the XOR of these t subfiles which are useful for the remaining t users in $\mathbf{U} \setminus \{u_i\}$ by transmitting

$$\bigoplus_{u_k \in \mathbf{U} \setminus \{u_i\}} \Gamma_{r_k, \mathbf{U} \setminus \{u_k\}},$$

where, $\Gamma_{r_k, \mathbf{U} \setminus \{u_k\}}$ is a subfile of the file r_k which is requested by u_k and is stored in the cache memory of all users in $\mathbf{U} \setminus \{u_k\}$ but is missing in u_k . Each user in $\mathbf{U} \setminus \{u_i\}$ can extract its required subfile using the side information in its cache.

- 2) Next, $\mathcal{S} \neq \emptyset$ and $|\mathcal{S}| < |\mathcal{T}|$. For each non-selfish user in \mathcal{T} , it just broadcasts the same packet as in Case 1. For each selfish user $u_s \in \mathcal{S}$, we pick an arbitrary user $u_t \in \mathcal{T}$ to transmit on behalf of u_s . According to the delivery procedure in Case 1, u_s should transmit a packet in the form of

$$\bigoplus_{u_k \in \mathbf{U} \setminus \{u_s\}} \Gamma_{r_k, \mathbf{U} \setminus \{u_k\}},$$

where $\Gamma_{r_k, \mathbf{U} \setminus \{u_k\}}$ is a subfile of file r_k requested by user u_k which is cached in all users in $\mathbf{U} \setminus \{u_k\}$ but is missing

in user u_k 's cache memory. Since u_s is selfish and does not participate in the delivery process, a user $u_t \in \mathcal{T}$ is picked to transmit

$$\bigoplus_{u_k \in \mathbf{U} \setminus \{u_s, u_t\}} \Gamma_{r_k, \mathbf{U} \setminus \{u_k\}},$$

which means that user u_t transmits the XOR of the same subfiles as u_s except the one which is useful for user u_t itself. In order to compensate for all the selfish users, a set of $|\mathcal{S}|$ arbitrary helpers are picked from \mathcal{T} , which is denoted by \mathcal{H} . Each helper in \mathcal{H} transmits in the above way on behalf of a user in \mathcal{S} . Note that each helper misses one subfile from its requested file which is stored in the cache memory of the selfish user it transmits on behalf of. In order to compensate for these missing subfiles, a user is picked from $\mathcal{T} \setminus \mathcal{H}$ to transmit the XOR of these $|\mathcal{S}|$ missing subfiles. Note that such a user can always be found since $|\mathcal{S}| < |\mathcal{T}|$. Each user in \mathcal{H} can then extract its missing subfile using the side information in its cache.

- 3) Last, $\mathcal{S} \neq \emptyset$ and $|\mathcal{S}| \geq |\mathcal{T}|$. For each non-selfish user in \mathcal{T} , it just broadcasts the same packet as in Case 1. By assumption, $|\mathbf{U}| = t+1$ and $|\mathcal{S}| \leq t-1$, which implies $|\mathcal{T}| \geq 2$. We let all users in \mathcal{T} be the helpers of $|\mathcal{T}|-1$ selfish users in \mathcal{S} . The delivery procedure is the same as in Case 2. Afterwards, we remove the $|\mathcal{T}|-1$ selfish users from \mathcal{S} to obtain \mathcal{S}' . Either the procedure in Case 2 or in Case 3 is repeated depending on whether $|\mathcal{S}'|$ is smaller than $|\mathcal{T}|$ or not. The delivery procedure ends when the transmissions of all selfish users have been compensated for.

Example 2. Consider the network in Example 1 where $S = 2$ of the users are selfish. Let the request vector be $\mathcal{R} = (\omega_1, \omega_2, \omega_3, \omega_4)$. There exists only one subset of $t+1 = 4$ users which is $\mathbf{U} = \{u_1, u_2, u_3, u_4\}$. Without loss of generality, let $\mathcal{S} = \{u_1, u_2\}$ be the set of selfish users. Therefore, $|\mathcal{S}| = |\mathcal{T}|$ where the delivery procedure lays on Case 3. Therefore, non-selfish users $\mathcal{T} = \{u_3, u_4\}$ multicast $\omega_{1,11} \oplus \omega_{2,8} \oplus \omega_{4,3}$ and $\omega_{1,12} \oplus \omega_{2,9} \oplus \omega_{3,6}$ which are useful for users $\{u_1, u_2, u_4\}$ and $\{u_1, u_2, u_3\}$, respectively. In order to compensate for selfish user u_1 , user u_3 multicasts $\omega_{2,7} \oplus \omega_{4,1}$ which is useful for users $\{u_2, u_4\}$ and u_4 unicasts $\omega_{3,4}$ which is useful for user u_3 . Moreover, in order to compensate for selfish user u_2 , user u_3 multicasts $\omega_{1,10} \oplus \omega_{4,2}$ which is useful for users $\{u_1, u_4\}$ and u_4 unicasts $\omega_{3,5}$ which is useful for user u_3 .

Theorem 1. Let M , N , K , and S be the cache size of each user, number of the popular files, number of users, and number of the selfish users in the network, respectively. For $t = \frac{MK}{N} \in \mathbb{Z}^+$, the following rate is achievable:

$$R(M) = \frac{1}{t \binom{K}{t}} \sum_{i=0}^S \binom{S}{i} \binom{K-S}{t+1-i} \left(t+1 + \left\lceil \frac{i}{t-i} \right\rceil \right) \quad (1)$$

Proof. The number of $(t+1)$ -subsets of \mathbf{U} which contain i selfish users is given by $\binom{S}{i} \binom{K-S}{t+1-i}$. Consider the number of

transmissions in such a subset. Since each non-selfish user transmits according to Case 1, the $t + 1 - i$ non-selfish users will first transmit a total of $t + 1 - i$ packets. Next, they need to compensate for the non-participation of the selfish users. Let q and r be the quotient and remainder when dividing i by $t - i$. Then the rule in Case 3 will be repeatedly applied for q times, which require a total of $q(t + 1 - i)$ transmissions. If $r = 0$, the delivery phase ends; otherwise, the rule in Case 2 will be applied, which requires $r + 1$ transmissions. Let $I(r > 0)$ be the indicator function which equals 1 if $r > 0$ and equals 0 otherwise. Consequently, the total number of transmissions by this subset of users is

$$\begin{aligned} & [t + 1 - i] + [q(t + 1 - i) + r + I(r > 0)] \\ &= t + 1 + q + I(r > 0) \\ &= t + 1 + \left\lceil \frac{i}{t - i} \right\rceil \end{aligned}$$

Since the length of each transmission is $\frac{1}{\binom{K}{t}}$ of the file size, the statement then follows. \square

IV. RANDOM CACHING

Due to the dynamic nature of the wireless networks, the deterministic caching policy may not be able to fully advantage D2D networks proposed in this paper. The deterministic caching approach, as explained in Section III, requires full control on the caching phase since some predetermined subfiles of each file must be placed in the cache memory of each user. In a wireless D2D network, however, devices may leave or join the network at any time. Therefore, a decentralized or random caching policy is more realistic and practical as it fits better these networks characteristics.

In contrast to the decentralized caching algorithm proposed in [3], randomly caching the bits of each file in D2D networks does not guarantee a successful recovery of the requested files due to the lack of a server (or base station) during the delivery phase to provide the subfiles that are not cached in any user. In order to overcome this issue, we follow the caching procedure in [7], where an MDS-code is employed to encode the library files before caching by the users. Consequently, any user can retrieve its requested file by having access to a subset of sufficient encoded symbols.

• **Caching phase:** Every file $\omega_1, \dots, \omega_N$ of length B bits is divided into I subfiles each of size $\frac{B}{I}$ bits. Each subfile is considered as a symbol in $\mathbb{F}_{2^{B/I}}$. Then, each file is encoded using an $(I, \frac{I}{r})$ MDS-code over $\mathbb{F}_{2^{B/I}}$, where r is the appropriate code rate which we will establish its necessary condition. Each user $u \in \mathcal{U}$ then randomly selects a set of $\frac{MI}{N}$ indices which indicate the corresponding encoded symbols of each file to be cached in the memory of user u .

By the end of the caching phase, the encoded symbols of any file ω_i for $i = 1, \dots, N$ can be partitioned as $\omega_{i,\mathcal{P}}$ where $\mathcal{P} \subset \{1, \dots, K\}$ such that $\omega_{i,\mathcal{P}}$ identifies the encoded symbols of file ω_i cached *exclusively* in the users of set \mathcal{P} . For instance, for a network with $N = 4$ files, $\omega_{1,\{2,4\}}$ specifies the encoded symbols of file ω_1 cached in users u_2 and u_4 , but *not* cached in

any other user. Similarly, $\omega_{1,\emptyset}$ specifies the encoded symbols of file ω_1 which are cached in no user. Let $\Gamma_{\omega_i,\mathcal{P}}$ be the block (i.e., concatenation) of encoded symbols of file ω_i specified by $\omega_{i,\mathcal{P}}$. As a consequence, the cache memory content of any user $u \in \mathcal{U}$ can be presented as $\Gamma_{\omega_1,\mathcal{P}}, \Gamma_{\omega_2,\mathcal{P}}, \dots, \Gamma_{\omega_N,\mathcal{P}}$ for all $\mathcal{P} \subset \mathcal{U}$ and $u \in \mathcal{P}$.

During the caching phase, each user caches $\frac{MI}{N}$ out of $\frac{I}{r}$ encoded symbols of each file. Therefore, the probability that a given encoded symbol of any file is cached in an arbitrary user is given by Mr/N . According to the Law of Large Numbers, the number of encoded symbols of any file ω_i exclusively cached by a set of users \mathcal{P} is given by

$$|\omega_{i,\mathcal{P}}| \approx \left(\frac{Mr}{N}\right)^{|\mathcal{P}|} \left(1 - \frac{Mr}{N}\right)^{K-|\mathcal{P}|} \frac{I}{r}, \quad (2)$$

with a high probability for large number of symbols I .

The delivery phase we are going to propose for D2D networks with selfish users guarantees that each user u at the end of the procedure has access to the encoded symbols of its requested file cached in all non-selfish users. Obviously, if there exist no selfish user in the network, each user can have access to all encoded symbols of its requested file which are cached in all other users [7].

Example 3. Consider a network with a file library of $N = 4$ files and $K = 4$ users each of which has a cache size $M = 2$. Assume user u_1 requests file ω_1 . As mentioned earlier, the cached symbols of encoded file ω_1 in user u_1 can be presented as $\Gamma_{\omega_1,\{1\}}, \Gamma_{\omega_1,\{1,2\}}, \Gamma_{\omega_1,\{1,3\}}, \Gamma_{\omega_1,\{1,4\}}, \Gamma_{\omega_1,\{1,2,3\}}, \Gamma_{\omega_1,\{1,2,4\}}, \Gamma_{\omega_1,\{1,3,4\}}$, and $\Gamma_{\omega_1,\{1,2,3,4\}}$. At the end of the delivery phase, user u_1 will also have access to the encoded symbols of file ω_1 cached in all of the other users (due to the fully cooperative network), namely $\Gamma_{\omega_1,\{2\}}, \Gamma_{\omega_1,\{3\}}, \Gamma_{\omega_1,\{4\}}, \Gamma_{\omega_1,\{2,3\}}, \Gamma_{\omega_1,\{2,4\}}, \Gamma_{\omega_1,\{3,4\}}$, and $\Gamma_{\omega_1,\{2,3,4\}}$. In other words, all of the encoded symbols of file ω_1 cached in all subsets $\mathcal{P} \subset \{1, 2, 3, 4\}$ of the users are known by user u_1 at the end of the delivery phase.

A. Rate of the MDS Code

Now we initiate a discussion on the rate of the MDS-code which is used in the caching phase.

Proposition 1. For a network with K users each of cache size M and a file library of size N , the number of the known encoded symbols of any requested file at each user by the end of the delivery phase is

$$\frac{I}{r} \sum_{i=1}^K \binom{K}{i} \left(\frac{Mr}{N}\right)^i \left(1 - \frac{Mr}{N}\right)^{K-i}. \quad (3)$$

Proof. Each user has access to all encoded symbols of its requested file cached in all subset of users $\mathcal{P} \subset \mathcal{U}$. The statement follows from (2) and the fact that there are $\binom{K}{i}$ subsets of size i . \square

Theorem 2. For a network with K users each of cache size M and a file library of size N where $\frac{MK}{N} > 1$, any arbitrary request vector $\mathcal{R} = (r_1, r_2, \dots, r_K)$ is recoverable by the users if all the files of the library are encoded using an MDS-code

with rate r such that $r < 1$ is the real positive root of the polynomial

$$f(r) \triangleq \sum_{i=0}^{K-1} \binom{K}{i} \left(-\frac{Mr}{N}\right)^{K-i} r^{K-i-1} + 1 = 0. \quad (4)$$

Proof. Since the I symbols (i.e., subfiles) of each file are encoded by an $(I, \frac{I}{r})$ MDS-code, any I symbols of the encoded file is sufficient to recover the requested file. Therefore, the number of symbols of the requested file known by the requesting user in (3) must satisfy

$$\frac{I}{r} \sum_{i=1}^K \binom{K}{i} \left(\frac{Mr}{N}\right)^i \left(1 - \frac{Mr}{N}\right)^{K-i} \geq I, \quad (5)$$

which can be re-written as

$$\frac{1}{r} \left[\left(1 - \frac{Mr}{N}\right)^K - 1 \right] + 1 = f(r) \leq 0. \quad (6)$$

Since $f(1) > 0$ and $f(0) = 1 - \frac{MK}{N} < 0$, the polynomial equation $f(r) = 0$ must have a real root $r \in (0, 1)$. \square

Note that the condition $\frac{MK}{N} > 1$ is required so that the total memory size of the K users is larger than the size of the whole library of files.

Example 4. Consider the same network as in Example 3. Then, the polynomial in (4) is given by

$$(1/2)^4 r^3 - 4(1/2)^3 r^2 + 6(1/2)^2 r - 4(1/2) + 1 = 0.$$

The real positive root of the polynomial is $r = 0.91$ which specifies the rate of the MDS-code to encode the library files. Encoding the library files with this rate guarantees that any arbitrary request vector is recoverable by the users.

Theorem 3. For a network with K users each with a cache of size M such that S out of K users are selfish, and a file library of size N where $\frac{M(K-S)}{N} > 1$, any arbitrary request vector $\mathcal{R} = (r_1, r_2, \dots, r_K)$ is recoverable by the users if all the files of the library are encoded using an MDS-code with rate r such that $r < 1$ is the real positive root of the polynomial

$$\sum_{i=0}^{K-S-1} \binom{K-S}{i} \left(-\frac{M}{N}\right)^{K-S-i} r^{K-S-1-i} + 1 = 0. \quad (7)$$

Proof. Note that Theorem 2 is true because each user is able to access the cache of $K-1$ other users at the end of the delivery phase. When S out of the K users are selfish, each selfish user can access the cache of $K-S$ non-selfish users while each non-selfish user can access the cache of $K-S-1$ other non-selfish users, so the request of a non-selfish user is harder to satisfy. The statement then follows directly from Theorem 2 by replacing K by $K-S$. \square

B. Delivery phase and the achievable rate

Now, we propose the delivery algorithm of our D2D random caching scheme where a subset of users in the network are selfish. Before describing the procedure, we need to define some notations first. Let $\Gamma_{\omega_i, \mathcal{P}} \oplus \Gamma_{\omega_i, \mathcal{P}'}$ be the XOR of these

two symbol blocks. In general, these two symbol blocks may not be of equal length. In that case, we zero-pad the shorter one so that the two blocks are of the same length. The notation can be generalized to more than two symbol blocks. Besides, in our delivery procedure, a symbol block is usually divided into a certain number, say f , of disjoint, equal-length segments. We call such a segment an $\frac{1}{f}$ -segment of a symbol block. Consider the case where there are several users, each of which has a symbol block. These symbol blocks are of the same length but their contents can be different. Each user transmits an $\frac{1}{f}$ -segment of its own symbol block. If the offsets of these $\frac{1}{f}$ -segments are all different, then we say that those segments are *disjoint*.

• **Delivery phase:** In each iteration of the delivery procedure, a subset $\mathbf{U} \subset \mathcal{U}$ of size $|\mathbf{U}| = u$ for all $u = K, K-1, \dots, 2$ is selected. Let $\mathcal{S} \subset \mathbf{U}$ and $\mathcal{T} \subset \mathbf{U}$ be the subsets of selfish and non-selfish users in \mathbf{U} , and let $|\mathcal{T}| = t$. Do the following for each \mathbf{U} that contains one or more non-selfish users:

- 1) If $|\mathcal{T}| = 1$, then let the only user $u \in \mathcal{T}$ transmits $\bigoplus_{v \in \mathbf{U}, v \neq u} \Gamma_{\omega_v, \mathbf{U} \setminus \{v\}}$.
- 2) If $|\mathcal{T}| \geq 2$, then
 - Pick an arbitrary user u^* from \mathcal{T} . User u^* transmits a disjoint $\frac{1}{t-1}$ -segment of $\bigoplus_{u \in \mathcal{T}, u \neq u^*} \Gamma_{\omega_u, \mathbf{U} \setminus \{u\}}$.
 - Each user $u \in \mathcal{T} \setminus \{u^*\}$ transmits a disjoint $\frac{1}{t-1}$ -segment of $\bigoplus_{v \in \mathbf{U}, v \neq u} \Gamma_{\omega_v, \mathbf{U} \setminus \{v\}}$.

Now, we establish the achievable rate based on the delivery algorithm proposed earlier.

Theorem 4. For a network with K users each of cache size M such that S out of K users are selfish, and a file library of size N which are encoded by a MDS-code of rate r satisfying the condition in Theorem 3, the following rate is achievable:

$$R(M) = \frac{1}{r} \sum_{i=2}^K R(i) \left(\frac{Mr}{N}\right)^{i-1} \left(1 - \frac{Mr}{N}\right)^{K-i+1}$$

where $R(i)$ is given by

$$R(i) = \sum_{j=0}^{i-2} \binom{S}{j} \binom{K-S}{i-j} \frac{i-j}{i-j-1} + (K-S) \binom{S}{i-1}. \quad (8)$$

Proof. The function $R(i)$ in (8) counts the number of transmissions for each subset of i users. In the first term of (8), $\binom{S}{j} \binom{K-S}{i-j}$ is the number of the subsets of i users which contain $i-j \geq 1$ non-selfish users. Each of such subsets transmits $\frac{1}{i-j-1}$ -segment of $i-j$ packets according to the Case 2 of the delivery phase. In the second term of (8), $(K-S) \binom{S}{i-1}$ is the number of the subsets of i users which contain only one non-selfish user and each of which transmits one packet according to the Case 1 of the delivery phase. Moreover, the number of the encoded symbols which are transmitted by each non-selfish user in a subset of i users is $\frac{1}{r} \left(\frac{Mr}{N}\right)^{i-1} \left(1 - \frac{Mr}{N}\right)^{K-i+1}$. Since the size of each symbol is $\frac{B}{T}$ bits, then the theorem follows. \square

Example 5. Consider a network with $K = 5$ users each of which has a cache memory of size $M = 2$ while $S = 2$ users are selfish, and a library of size $N = 5$ files. In order to satisfy any arbitrary request, all files must be encoded by a MDS-code of rate $r = 0.44$ follows from (7) in Theorem 3. Assume, without loss of generality that users u_1 and u_2 are selfish and the request vector is $\mathcal{R} = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$. Then, the delivery phase will be carried out for all subsets $U \subset \mathcal{U}$ of size $|U| = 5, 4, 3, 2$. Suppose, $U = \{1, 2, 3, 4, 5\}$ which contains three non-selfish users $\mathcal{T} = \{3, 4, 5\}$. Following the Case 2 of the delivery phase, we choose $u^* = u_3$. Then, u_3 will transmit $\frac{1}{2}\Gamma_{\omega_4, \{1,2,3,5\}} \oplus \Gamma_{\omega_5, \{1,2,3,4\}}$. Moreover, users u_4 and u_5 will respectively transmit $\frac{1}{2}\Gamma_{\omega_1, \{2,3,4,5\}} \oplus \Gamma_{\omega_2, \{1,3,4,5\}} \oplus \Gamma_{\omega_3, \{1,2,4,5\}} \oplus \Gamma_{\omega_5, \{1,2,3,4\}}$ and $\frac{1}{2}\Gamma_{\omega_1, \{2,3,4,5\}} \oplus \Gamma_{\omega_2, \{1,3,4,5\}} \oplus \Gamma_{\omega_3, \{1,2,4,5\}} \oplus \Gamma_{\omega_4, \{1,2,3,5\}}$. Now, let $U = \{u_1, u_2, u_3\}$ which contains only one non-selfish user $\mathcal{T} = \{u_3\}$. Following the Case 1 of the delivery phase, user u_3 will transmit $\Gamma_{\omega_1, \{2,3\}} \oplus \Gamma_{\omega_2, \{1,3\}}$. The same procedure will be carried out for all other U . At the end of the delivery phase, all users will have access to all encoded symbols cached in the network except the ones cached in selfish users. The transmission rate for this network setting is $R(2) = 3.05$ following from Theorem 4.

The performance of the “deterministic” and “random” caching schemes in a partially cooperative D2D network is depicted in Figure 2 for two different network settings. Figure 2a and 2b compare the achievable rates of deterministic and random caching schemes for a network with parameters $K = 100$ users, $S = 10$ selfish users, $N = 50$ files and $K = 50$ users, $S = 10$ selfish users, $N = 100$ files, respectively. The simulation results show that although the deterministic scheme has a slightly better performance (i.e., smaller transmission rate), the gap vanishes as the cache memory size M increases. The lower bounds depicted in Figure 2 is based on the cut-set argument [7] and is given by

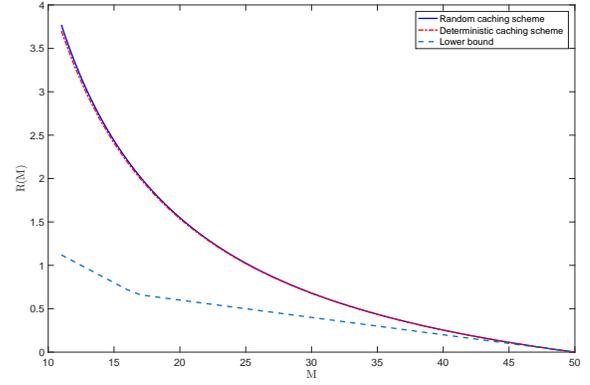
$$R^*(M) \geq \max_{l=\{1,2,\dots,\min\{K,N\}\}} \left(l - \left\lfloor \frac{l}{T} \right\rfloor M \right),$$

where, $R^*(M)$ is the optimal achievable rate defined as

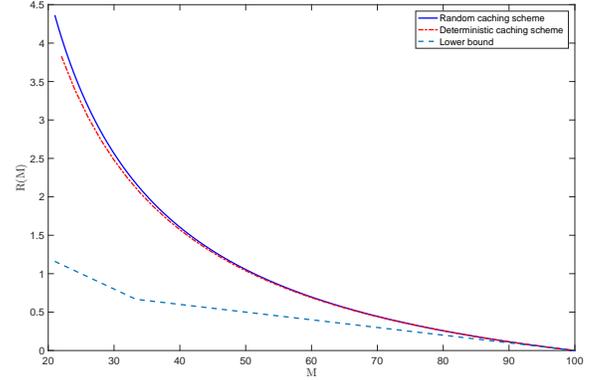
$$R^*(M) \triangleq \inf \{R(M) : R(M) \text{ is achievable}\}.$$

V. CONCLUSIONS

In this paper, we consider partially cooperative D2D caching networks where a group of users are selfish. All users cache the file library during the caching phase while only non-selfish users transmit their useful cache content to other users. We propose a deterministic caching scheme where the users’ cache content is a deterministic function of the file library. We also propose a more realistic decentralized caching scheme where users randomly cache some parts of the files. We show that any arbitrary user request is recoverable under our proposed caching schemes at the presence of selfish users in the D2D networks.



(a) $K = 100$, $N = 50$, $S = 20$.



(b) $K = 50$, $N = 100$, $S = 10$.

Fig. 2: Performance of the deterministic and random caching schemes in a partially cooperative D2D networks with S selfish users out of K users and a file library of size N files.

REFERENCES

- [1] CISCO. (2017, Feb.) Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021.
- [2] M. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Trans. Info. Theory*, vol. 60, no. 5, pp. 2856–67, May 2014.
- [3] M. A. Maddah-Ali and U. Niesen, “Decentralized coded caching attains order-optimal memory-rate tradeoff,” *IEEE/ACM TRANSACTIONS ON NETWORKING*, VOL. 23, NO. 4., vol. 23, no. 4, pp. 1029–1040, Aug. 2015.
- [4] N. Golrezaei, M. Ji, A. F. Molisch, A. G. Dimakis, and G. Caire, “Device-to-device communications for wireless video delivery,” in *Proc. Conf. Rec. 46th Asilomar Conf. Signals, Syst. Comput. (ASILOMAR)*, Nov. 2012, pp. 930–933.
- [5] M. Ji, G. Caire, and A. F. Molisch, “Optimal throughput-outage trade-off in wireless one-hop caching networks,” in *proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 1461–1465.
- [6] 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.
- [7] M. Ji, G. Caire, and A. F. Molisch, “Fundamental limits of caching in wireless d2d networks,” *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6833–6859, 2015.
- [8] —, “Wireless device-to-device caching networks: Basic principles and system performance,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, Jan. 2016.