Improved Delivery Rate-Cache Capacity Trade-off For Centralized Coded Caching

Mohammad Mohammadi Amiri
Electrical and Electronic Engineering Department
Imperial College London
London, SW7 2BT
Email: m.mohammadi-amiri15@imperial.ac.uk

Deniz Gündüz
Electrical and Electronic Engineering Department
Imperial College London
London, SW7 2BT
Email: d.gunduz@imperial.ac.uk

***Abstract*—*Centralized coded caching* problem, in which a server with N distinct files, each with the size of F bits, serves K users, each equipped with a cache of capacity MF bits, is considered. The server is allowed to proactively cache contents into user terminals during the *placement phase*, without knowing the particular user requests. After the placement phase, each user requests one of the N files from the server, and all the users' requests are satisfied simultaneously by the server through an error-free shared link during the *delivery phase*. A novel coded caching algorithm is proposed, which is shown to achieve a smaller delivery rate compared to the existing coded caching schemes in the literature for a range of N and K values; particularly when the number of files is larger than the number of users in the system.**

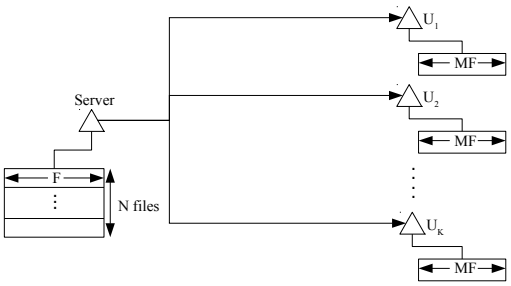

Fig. 1. Caching system consisting of one server storing N popular files, each with size F bits, and K users, each with a cache of capacity MF bits.

I. Introduction

Increasing number of users and their growing demand for high data rate content lead to network congestion, particularly during peak hours. Exploiting low-cost storage at user terminals, and utilizing the channel resources during off-peak hours can be an effective way to combat this problem, and to reduce the peak data traffic [1], [2]. Popular contents can be stored at users' caches during off-peak hours *proactively*, shifting part of the network traffic to off-peak hours [3].

In classical *uncoded caching*, popular contents are delivered, either in part or fully, to users proactively, which reduces the traffic during peak periods. This leads to a *local caching gain* [4], which is limited by the cache size of each user. Recently it has been shown that *coded caching* can offer significant advantages over uncoded caching in certain scenarios; particularly when the users are served over a common shared channel during the peak traffic period [1]. Maddah-Ali and Niesen proposed a centralized coded caching scheme, and showed that it provides a *global caching gain* by creating and exploiting coded multicasting opportunities.

In the *centralized setting*, also considered in this paper, the number and identity of users that participate in the caching scheme are assumed to be known in advance. While the proposed scheme in [1] can be shown to achieve the optimal performance when the normalized cache capacity M (normalized by the size of the files) satisfies $M \geq N(1-1/K)$, i.e., for large cache capacities, the optimal delivery rate-cache capacity trade-off is still not fully characterized. An improved centralized coded caching scheme (also considered in [1] for the special case of two files and two users) is proposed in [5] when the number of users is not less than the number of files, i.e., $N \leq K$. It is shown that this scheme is optimal when $M \leq 1/K$, i.e., for small cache sizes. The scheme in [5] can be considered as the dual of the one proposed in [1]; while the latter caches portions of the files and sends their XOR-ed versions during the delivery phase, the former directly caches the XOR-ed contents. Other centralized caching schemes have also been proposed recently. The scheme in [6] is designed for the case of $N = 2$ files, and achieves a lower delivery rate when $1/K < M < 1$. By exploiting multicasting opportunities across users with the same demand, the authors in [7] present a new coded caching scheme for the case $N < K$, which improves upon the state of the art also for $N = 2$ files. In addition to the cut-set lower bound on the delivery rate derived in [1], a tighter lower bound is proposed in [8].

In this paper, we propose yet another novel centralized coded caching algorithm, which brings together ideas from both caching schemes introduced in [1] and [5]. This new caching algorithm is introduced by considering a normalized cache capacity of $M = (N-1)/K$. For this cache capacity, we show that the proposed algorithm achieves a lower delivery rate compared to all the existing schemes in the literature, as well as their convex combinations through memory-sharing, when N and K have a common divisor $c > 1$, and satisfy $4 \leq N < K \leq 3N/2$. Through memory-sharing arguments, the improvement in the delivery rate can be extended to a larger set of cache capacities satisfying $1/K \leq M \leq N/K$

when $K < 3N/2$, and $1/K \leq M \leq 2N/K$ when $K = 3N/2$.

The rest of this paper is organized as follows. In Section II, we state the system model and the relevant previous results. The novel coded caching scheme is first introduced for $c = 2$ in Section III, and then extended to any $c \geq 2$ in Section IV. In Section V, we present numerical results demonstrating the gains of the proposed caching algorithm, and our conclusions are summarized in Section VI.

II. System Model and Previous Results

A. System Model

Following the model introduced in [1], we consider a centralized coded caching system as depicted in Fig. 1. There are K users $U_1, ..., U_K$, and N files $W_1, ..., W_N$ distributed uniformly across $\left\{1, ..., 2^F\right\}$, in the system. All the files have the same size F bits, and each user is equipped with a cache of capacity MF bits.

There are two phases of data transmission from the server to the users. In the initial *placement phase*, the server fills in the limited cache memories without knowing the requests of the users. The user demands $d_1, ..., d_K$, where $d_i \in \{W_1, ..., W_N\}$, are revealed to the server and the users after the placement phase. In the following *delivery phase*, all the users' requests must be satisfied from the local caches together with the data transmitted by the server over the shared link. The goal is to design the placement and delivery phases jointly in order to reduce the rate required in the delivery phase, such that any demand combination can be satisfied, i.e., each user can decode the requested file with arbitrarily low probability of error for sufficiently large F.

The *delivery rate*, R, of a coded caching scheme is the maximum of the rates transmitted in the delivery phase to satisfy all the user demands, maximum taken over all possible demand combinations, i.e., the delivery rate corresponding to the worst-case user demands. There is a trade-off between the cache capacity, M, and the corresponding delivery rate, R. For given K and N, the *delivery rate-cache capacity trade-off*, $R^*(M)$, is defined as the minimum delivery rate to satisfy all possible user demands for a normalized cache capacity of M. We refer the reader to [9] for a more rigorous description of the system model.

B. Previous Results

For $1 < N \leq K$, let $R_b(M)$ denote the best known delivery rate-cache capacity trade-off for centralized coded caching in the literature. For $N \leq K < 3N/2$, the achievable rates for $M = 1/K$ and $M = N/K$ are $R_b(1/K) = N - N/K$ and $R_b(N/K) = (K-1)/2$, proposed in [5] and [1], respectively, and the best delivery rate for $1/K \leq M \leq N/K$ is obtained by the convex hull of these two trade-off curves through memory-sharing [1]. Consequently, for $N \leq K < 3N/2$, we have

$$R_b(M) = \frac{(K-1)}{2(N-1)}\left[(K-2N)M + \frac{2N^2}{K} - 1\right], \quad \text{if } \tfrac{1}{K} \leq M \leq \tfrac{N}{K}. \quad (1)$$

On the other hand, for $3N/2 \leq K \leq 2N$, it can be shown that the best delivery rate is achieved by memory-sharing between the schemes for points $M = 1/K$ and $M = 2N/K$, proposed in [5] and [1], respectively. In this case, we have $R_b(1/K) = N - N/K$ and $R_b(2N/K) = (K-2)/3$. As a result, for $3N/2 \leq K \leq 2N$, we have

$$R_b(M) = \frac{K^2 - (3N+2)K + 3N}{3(2N-1)}M + \frac{-K^2 + \left(6N^2+2\right)K - 6N^2}{3K(2N-1)}, \quad \text{if } \tfrac{1}{K} \leq M \leq \tfrac{2N}{K}. \quad (2)$$

In our setting, we assume that N and K have a common divisor $c > 1$, and $K = N + ct$, for some $t \in \mathcal{Z}^+$, such that $1 \leq t \leq N/2c$ and $N \geq 4$, where $\mathcal{Z}^+$ is the set of positive integers. We consider a cache capacity of $M = (N-1)/K$. In the following, we first present the placement and delivery phases for the proposed caching algorithm. We then show that it has a lower delivery rate compared to the state of the art. We will then characterize an improved cache capacity-delivery rate trade-off by combining the proposed caching scheme with the existing ones through memory-sharing.

III. Proposed Coded Caching Scheme For $c = 2$

We first consider the case when both N and K are even numbers, i.e., $c = 2$. Although the server has no information about the users' demands in the placement phase, the content of the caches has a huge impact on the data that needs to be served in the delivery phase after the user demands are revealed. Hence, the placement phase has a significant role in reducing the required delivery rate, and the two phases need to be designed jointly.

A. Placement phase

For cache capacity $M = (N-1)/K$, we split each file into K non-overlapping distinct subfiles $W_{i,j}$, each of length F/K bits, and place $W_{i,j} \oplus W_{i+1,j}$ in the cache of j-th user, for $i = 1, ..., N-1$ and $j = 1, ..., K$, where $\oplus$ denotes the bitwise XOR operation[1]. Therefore, we have placed $(W_{1,j} \oplus W_{2,j}, ..., W_{N-1,j} \oplus W_{N,j})$ in the j-th user's cache. Note that, each subfile of the files is cached exactly in the cache of one user (in XOR-ed form), and there is a symmetry among the cache contents.

We will consider the case $N = 8$, $K = 12$ and $M = 7/12$ as a running example to clarify the main techniques of our coded caching scheme. In this scenario, coded contents, $(W_{1,j} \oplus W_{2,j}, W_{2,j} \oplus W_{3,j}, W_{3,j} \oplus W_{4,j}, W_{4,j} \oplus W_{5,j}, W_{5,j} \oplus W_{6,j}, W_{6,j} \oplus W_{7,j}, W_{7,j} \oplus W_{8,j})$, are placed in the cache of j-th user during the placement phase.

We highlight that the placement phase of our caching algorithm combines the attributes of the two dual schemes proposed in [1] and [5]. We divide the contents into many smaller portions, and cache XOR-ed portions into each user's cache. However, as opposed to [5] we XOR only two subfiles. These portions are chosen carefully to create symmetry

[1]Note that we implicitly assume that $W_i = \left[W_{i,1}, ..., W_{i,K}\right]$ is the F-length binary representation of file i.

among user cache contents in order to maximize multicasting opportunities in the delivery phase.

B. Delivery phase

The delivery rate should be sufficient to satisfy any demand combination. To present the delivery phase, the user demands are assumed as distinct as possible; that is, N distinct files are requested by N users, and the remaining $2t$ users request $2t$ distinct files. Accordingly, by re-labeling the files, it is assumed that the user requests are as follows:

$$d_i = W_{i-N\lfloor \frac{i-1}{N} \rfloor}, \quad i = 1, ..., K, \tag{3}$$

where $\lfloor x \rfloor$ is the largest integer no greater than x. Since it is assumed that $t \leq N/4$, the requests of the first $2t$ users are repeated by the last $2t$ users, and the remaining $(N-2t)$ users have distinct non-repeating requests. For the example under consideration, we have $(d_1, ..., d_{12}) = (W_1, ..., W_8, W_1, ..., W_4)$.

We divide the messages transmitted in the delivery phase into different parts, and in the following, we explain the purpose of each part of the delivery phase in detail.

I. In the first part, the server transmits

$$W_{i,i+f(i)}, \quad i = 1, ..., N, \tag{4}$$

where $f : (\mathcal{Z}^+ \to \{-1, 1\})$ is defined as

$$f(i) \triangleq \begin{cases} 1, & \text{if } i \text{ is odd,} \\ -1, & \text{if } i \text{ is even.} \end{cases} \tag{5}$$

In this way, each user U_j, for $j = 1, ..., N$, can recover the subfiles $W_{1,j}, ..., W_{N,j}$ stored in its cache. Then, the subfiles

$$W_{i,i+N+f(i)}, \quad i = 1, ..., 2t, \tag{6}$$

are delivered by the server, and each of the remaining $2t$ users U_j, for $j = N+1, ..., K$, can retrieve all the subfiles $W_{1,j}, ..., W_{N,j}$.

In our example with $t = 2$, the server delivers the subfiles $W_{1,2}$, $W_{2,1}$, $W_{3,4}$, $W_{4,3}$, $W_{5,6}$, $W_{6,5}$, $W_{7,8}$, $W_{8,7}$, $W_{1,10}$, $W_{2,9}$, $W_{3,12}$, and $W_{4,11}$ in the first part of the delivery phase, which enable each user to recover all the subfiles placed in its cache in XOR-ed form. This amounts to a total delivery rate of 1.

II. In the second part, to satisfy the demands of the first $2t$ users, the server delivers

$$W_{i,j} \oplus W_{j-N\lfloor \frac{j-1}{N} \rfloor, i}, \; i = 1, ..., 2t; \tag{7}$$
$$j = i + \frac{3 + f(i)}{2}, ..., K; \; j \neq N + i, N + i + f(i).$$

Having access to subfile $W_{j-N\lfloor (j-1)/N \rfloor, i}$ locally (through the contents delivered in part I), each user U_i can obtain all the portions of its requested file, but the one placed in the cache of the user with the same demand, i.e., user U_{i+N}, for $i = 1, ..., 2t$ and $j = i + (3 + f(i))/2, ..., K \neq N + i, N + i + f(i)$. At the same time, each user U_k, for $2t < k \leq N$, can also retrieve the subfiles of its desired file which are in the cache of users $U_1, ..., U_{k-(3-f(k))/2}$, and for $k > N$, user U_k can decode the subfiles of its demand stored in the cache of users $U_1, ..., U_{k-(3-f(k))/2}$ excluding U_{k-N}.

In our example, the server delivers the following in the second phase of the delivery phase: $W_{1,3} \oplus W_{3,1}$, $W_{1,4} \oplus W_{4,1}$, $W_{1,5} \oplus W_{5,1}$, $W_{1,6} \oplus W_{6,1}$, $W_{1,7} \oplus W_{7,1}$, $W_{1,8} \oplus W_{8,1}$, $W_{1,11} \oplus W_{3,1}$, $W_{1,12} \oplus W_{4,1}$, $W_{2,3} \oplus W_{3,2}$, $W_{2,4} \oplus W_{4,2}$, $W_{2,5} \oplus W_{5,2}$, $W_{2,6} \oplus W_{6,2}$, $W_{2,7} \oplus W_{7,2}$, $W_{2,8} \oplus W_{8,2}$, $W_{2,11} \oplus W_{3,2}$, $W_{2,12} \oplus W_{4,2}$, $W_{3,5} \oplus W_{5,3}$, $W_{3,6} \oplus W_{6,3}$, $W_{3,7} \oplus W_{7,3}$, $W_{3,8} \oplus W_{8,3}$, $W_{3,9} \oplus W_{1,3}$, $W_{3,10} \oplus W_{2,3}$, $W_{4,5} \oplus W_{5,4}$, $W_{4,6} \oplus W_{6,4}$, $W_{4,7} \oplus W_{7,4}$, $W_{4,8} \oplus W_{8,4}$, $W_{4,9} \oplus W_{1,4}$, and $W_{4,10} \oplus W_{2,4}$. This corresponds to a rate of $28/12 = 7/3$ for the second part.

III. In the third part, the server satisfies the requests of users U_{2t+1} to U_N. This part can be handled similarly to the previous one, with the slight difference that the requests of users U_{2t+1} to U_N are not repeated by the other users. So, we need to send

$$W_{i,j} \oplus W_{j-N\lfloor \frac{j-1}{N} \rfloor, i}, \quad i = 2t+1, ..., N; \quad j = i + \frac{3 + f(i)}{2}, ..., K. \tag{8}$$

Receiving these bits, user U_i can decode all subfiles of its requested file placed in the cache of users $U_{i+(3+f(i))/2}, ..., U_K$, for $i = 2t+1, ..., N$. Thus, receiving these bits together with the bits delivered in parts I and II enables users $U_{2t+1}, ..., U_N$ to obtain their desired files.

In our example, $W_{5,7} \oplus W_{7,5}$, $W_{5,8} \oplus W_{8,5}$, $W_{5,9} \oplus W_{1,5}$, $W_{5,10} \oplus W_{2,5}$, $W_{5,11} \oplus W_{3,5}$, $W_{5,12} \oplus W_{4,5}$, $W_{6,7} \oplus W_{7,6}$, $W_{6,8} \oplus W_{8,6}$, $W_{6,9} \oplus W_{1,6}$, $W_{6,10} \oplus W_{2,6}$, $W_{6,11} \oplus W_{3,6}$, $W_{6,12} \oplus W_{4,6}$, $W_{7,9} \oplus W_{1,7}$, $W_{7,10} \oplus W_{2,7}$, $W_{7,11} \oplus W_{3,7}$, $W_{7,12} \oplus W_{4,7}$, $W_{8,9} \oplus W_{1,8}$, $W_{8,10} \oplus W_{2,8}$, $W_{8,11} \oplus W_{3,8}$, and $W_{8,12} \oplus W_{4,8}$ are delivered by the server, with a total rate of $20/12 = 5/3$ for this part.

IV. Now, consider the last $2t$ users requesting the same files as the first $2t$ users. The server has to send

$$W_{i,j} \oplus W_{j-N,i+N}, \quad i = 1, ..., 2t-2; \quad j = N + i + \frac{3 + f(i)}{2}, ..., K. \tag{9}$$

Having received these bits, the only portion remaining for user U_i to satisfy its demand is the one placed in the cache of user U_{i-N}, for $i = N+1, ..., K$. Note that, this happens only if $t \geq 2$, and in our example, $W_{1,11} \oplus W_{3,9}$, $W_{1,12} \oplus W_{4,9}$, $W_{2,11} \oplus W_{3,10}$, and $W_{2,12} \oplus W_{4,10}$ are transmitted in this part with a total rate of $4/12 = 1/3$.

V. Finally, to satisfy the demands of the users with the same requests, i.e., users U_k and U_{k+N}, for $k = 1, ..., 2t$, the following XOR-ed contents should be sent in the last part of the delivery phase:

$$W_{i,i} \oplus W_{i,i+N}, \quad i = 1, ..., 2t. \tag{10}$$

Observe that $W_{1,1} \oplus W_{1,9}$, $W_{2,2} \oplus W_{2,10}$, $W_{3,3} \oplus W_{3,11}$, and $W_{4,4} \oplus W_{4,12}$ need to be shared in our example. This

$$R_c(M) = \begin{cases} \frac{K^2-2K-2NK+4N}{2(N-2)}M + N - \frac{N}{K} - \frac{K^2-2K-2NK+4N}{2K(N-2)}, & \text{if } \frac{1}{K} \le M \le \frac{N-1}{K}, \\ \left(\frac{1}{2}K - N\right)M + \frac{1}{2}(K-N-1) + \frac{N^2}{K}, & \text{if } \frac{N-1}{K} < M \le \frac{N}{K},\ N < K < \frac{3N}{2}, \\ \frac{-K^2+2K-6N}{6(N+1)}M + \frac{K-2}{3} + \frac{N\left(K^2-2K+6N\right)}{3K(N+1)}, & \text{if } \frac{N-1}{K} < M \le \frac{2N}{K},\ K = \frac{3N}{2}. \end{cases} \tag{11}$$

leads to a delivery rate of $1/3$ for this last part.

After receiving all these bits, all the users' requests are satisfied. In particular, in our example, the server can satisfy all the requests by transmitting a total of $17F/3$ bits in the delivery phase, while this number for the best achievable scheme in the literature is $17.2F/3$. The delivery rate for the general case, and its comparison with the existing results in the literature are presented below.

C. Delivery Rate

The main result of this section, that is, the delivery rate achieved by the proposed coded caching algorithm, is stated in the following theorem, whose proof is skipped due to space limitations.

Theorem 1. *In a centralized caching system with N files and K users, where N and K are both even numbers satisfying $4 \le N < K \le 3N/2$, if each user has a cache of capacity $M = (N-1)/K$, the following delivery rate is achievable:*

$$R_c\left(\frac{N-1}{K}\right) = \frac{K}{2} + \frac{N}{K} - 1. \tag{12}$$

According to (1), for $M = (N-1)/K$ and $N \le K < 3N/2$, the best delivery rate achieved by memory-sharing between the schemes proposed in [1] and [5] is given by

$$R_b\left(\frac{N-1}{K}\right) = \frac{K}{2} - \frac{K^2 - (N+2)K + 2N}{2K(N-1)}. \tag{13}$$

Now we show that $R_c((N-1)/K) < R_b((N-1)/K)$. By substituting $(K-2t)$ instead of N, we have

$$R_c\left(\frac{N-1}{K}\right) = \frac{K}{2} - \frac{2t}{K}, \tag{14a}$$

$$R_b\left(\frac{N-1}{K}\right) = \frac{K}{2} - \frac{2t(K-2)}{K(2(K-2t)-2)}. \tag{14b}$$

All we need to show is $(K-2) < (2(K-2t)-2)$, which follows since $4t < K$. Now, consider $K = 3N/2$. Based on (2), the best known achievable delivery rate for $M = (N-1)/K$ is

$$R_b\left(\frac{N-1}{K}\right) = \frac{6K^2 - 4K + 3}{3(4K-3)} - \frac{1}{3}. \tag{15}$$

The achievable delivery rate of our scheme in this case is

$$R_c\left(\frac{N-1}{K}\right) = \frac{K}{2} - \frac{1}{3}, \tag{16}$$

which is again smaller than (15). As a result, the delivery rate of the proposed scheme for $M = (N-1)/K$, when $4 \le N < K \le 3N/2$, is less than that required by memory-sharing between the two schemes proposed in [1] and [5].

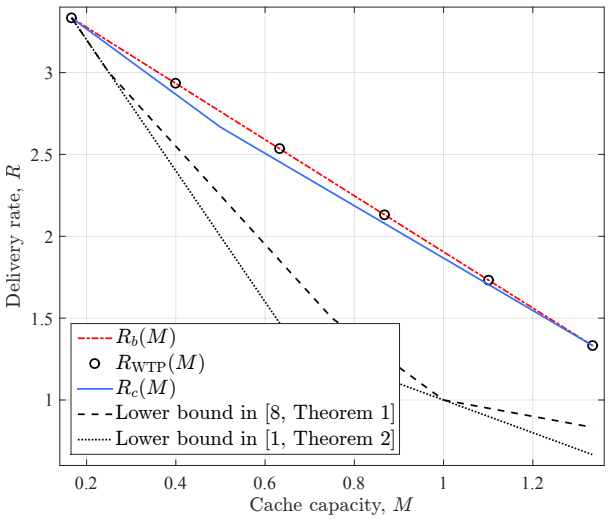

Fig. 2. Delivery rate comparison for $N = 4$ and $K = 6$, i.e., $c = 2$ and $t = 1$, when $1/K \le M \le 2N/K$. $R_b(M)$ can be achieved by memory-sharing between the schemes for $M = 1/K$ and $M = 2N/K$ proposed in [5] and [1], respectively.

Now, we can extend our result to the interval $1/K \le M \le N/K$ when $N < K < 3N/2$, and $1/K \le M \le 2N/K$ when $K = 3N/2$, by memory-sharing between our scheme and those presented in [1] and [5]. This is stated in the following corollary.

Corollary 1. *In a centralized coded caching system with N files and K users, where both N and K are even numbers such that $4 \le N < K \le 3N/2$, the delivery rate-cache capacity trade-off given in* (11) *at the top of this page is achievable.*

IV. The General Case ($c \ge 2$)

Building upon the placement phase presented in the previous section, the proposed coded delivery algorithm can be extended to the general case for any $c \ge 2$; and therefore, the improvement in the delivery rate extends to any N and K values, satisfying $N < K \le 3N/2$, as long as they are not relatively prime. We present the general result in the following theorem, whose proof be found in [9].

Theorem 2. *For K users, N files, and a cache capacity of $M = (N-1)/K$, if N and K have a common divisor $c \ge 2$, and satisfy $4 \le N < K \le 3N/2$, the delivery rate given by* (12) *is achievable in a centralized manner.*

Since the delivery rate of the case $c \ge 2$ is equal to that of $c = 2$ when $M = (N-1)/K$, the same procedures argued in the previous section can be followed upon to prove that $R_c((N-1)/K) < R_b((N-1)/K)$ for any $c > 1$. Thus,

the improved delivery rate-cache capacity trade-off given by (11) for $c = 2$ is also achievable for the case $c \geq 2$. Note that the generalized scheme extends the improved delivery rate to a larger set of N and K values.

Remark 1. *Denoting number of users requesting file W_i by K_i, for $i = 1, ..., N$, user demand combination specified in* (3) *corresponds to $1 \leq K_i \leq 2$, $\forall i$. It can be shown that by performing the proposed caching scheme for the setting considered in this paper, delivery rate* (12) *is sufficient to satisfy all user demands when either $K_i = 0$ or $K_i > 2$. Hence, the user demand combination in* (3) *can be considered as the worst-case.*

V. Simulation Results

In Fig. 2, we compare the delivery rate-cache capacity trade-off achieved by the proposed coded caching algorithm for the case $N = 4$ and $K = 6$, i.e., $c = 2$ and $t = 1$, when $M \in [1/K, 2N/K]$, with the best known achievable scheme in the literature. For this range of cache capacities, the delivery rate of the centralized coded caching scheme investigated in [7], denoted by $R_{\text{WTP}}(M)$, is also included in the figure. $R_{\text{WTP}}(M)$ achieves the same delivery rate as the best achievable scheme in the literature for this interval. It can be seen that the superiority of the proposed scheme for cache capacity $M = (N-1)/K$ leads to a lower delivery rate for the range of cache capacities under consideration compared to the existing schemes in the literature. We also consider the cut-set lower bound [1, Theorem 2], and the tightest known lower bound on the delivery rate derived in [8, Theorem 1]. The gap to the lower bound remains for most M values despite the improvement in the achievable rate.

In Fig. 3, for $M = (N-1)/K$, the delivery rate of our proposed scheme, $R_c((N-1)/K)$, is compared with the best achievable delivery rate in the literature, $R_b((N-1)/K)$, as a function of N when $K = 300$. Note that, the proposed scheme can improve upon the best known result for $2K/3 \leq N = cn < K$, such that $c \in \{2, 3, 5\}$ and $\forall n \in \mathcal{Z}^+$. We consider the case $c = 2$ in this figure since it corresponds to a larger range of N and K values. The delivery rate of the coded scheme proposed in [7], $R_{\text{WTP}}((N-1)/K)$, is equal to $R_b((N-1)/K)$ in this scenario, and thus, it is omitted in this figure. Interestingly, the superiority of our scheme is more pronounced for relatively small number of files in the above interval.

VI. Conclusion

We have proposed a novel centralized coded caching algorithm for a normalized cache capacity of $M = (N-1)/K$. The proposed coding algorithm combines the benefits of the known schemes in the literature proposed for $M = 1/K$ in [5] and for $M = N/K$ in [1]. Similarly to these schemes, we divide each file into smaller portions; however, instead of caching a single content obtained by XORing many portions of different files as in [5], we cache many components, each obtained by XORing only two portions of two different files. We have proved that if N and K have a common divisor greater than 1, i.e., N and K are not relatively prime, and satisfy $4 \leq N < K \leq 3N/2$, the proposed coded caching algorithm achieves a lower delivery rate than the known schemes in the literature. We have then extended this improvement to a larger range of cache capacities through memory-sharing between the proposed scheme and the known schemes in the literature.

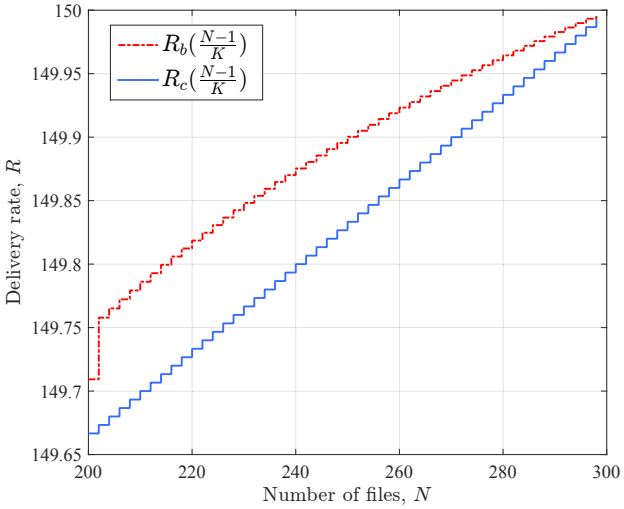

Fig. 3. Delivery rate for $K = 300$ and $N = 200, 202, ..., 298$, when $M = (N-1)/K$. The rate $R_b((N-1)/K)$ is calculated through (13) apart from the case $N = 200$ ($K = 3N/2$), which is determined by (15).

While the superiority of the delivery rate of our proposed scheme is relatively small, we remark that the total number of bits sent over the shared link in the delivery phase is scaled with F, which is assumed to be very high. The proposed scheme is expected to improve the rates in the decentralized setting as well, which is currently under consideration.

References

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inform. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[2] M. Gregori, J. Gomez-Vilardebo, J. Matamoros, and D. Gündüz, "Wireless content caching for small cell and D2D networks," *to appear, IEEE J. Sel. Areas Commun.*, 2016.

[3] K. C. Almeroth and M. H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 6, pp. 1110–1122, Aug. 1996.

[4] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, San Diego, CA, Mar. 2010, pp. 1–9.

[5] Z. Chen, P. Fan, and K. B. Letaief, "Fundamental limits of caching: Improved bounds for small buffer users," *arXiv: 1407.1935v2 [cs.IT]*, Jul. 2014.

[6] S. Sahraei and M. Gastpar, "K users caching two files: An improved achievable rate," *arXiv: 1512.06682 [cs.IT]*, Dec. 2015.

[7] K. Wan, D. Tuninetti, and P. Piantanida, "On caching with more users than files," *arXiv: 1601.063834v2 [cs.IT]*, Jan. 2016.

[8] A. Sengupta, R. Tandon, and T. C. Clancy, "Improved approximation of storage-rate tradeoff for caching via new outer bounds," in *Proc. IEEE Int'l Symp. on Inf. Theory*, Hong Kong, Jun. 2015, pp. 1691–1695.

[9] M. Mohammadi Amiri and D. Gündüz, "Fundamental limits of caching: Improved delivery rate-cache capacity trade-off," *[online]. Available. http://www.iis.ee.ic.ac.uk/dgunduz/Papers/Journal/AG_TC16.pdf.*