# A Coded Caching Scheme with Linear Sub-packetization and its Application to Multi-Access Coded Caching

Anjana A. Mahesh, and B. Sundar Rajan

Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru 560012, KA, India

E-mail: {anjanamahesh,bsrajan}@iisc.ac.in

*Abstract*—This paper addresses the problem of exponentially increasing sub-packetization with the number of users in a centralized coded caching system by introducing a new coded caching scheme inspired by the symmetric neighboring consecutive side information index coding problem. The scheme has a placement policy where the number of sub-packets required grows only linearly with the number of users, with no restriction on file size, and a delivery policy which is instantaneously decodable. Further, an application of the new delivery scheme in a multi-access coded caching set-up is studied and a few results in that direction are presented. In particular, in the multi-access set-up, for cases where optimality rate-memory trade-off characterizations are available, it is shown that the new delivery scheme achieves optimal or near-optimal rates.

*Index Terms*—Coded Caching, Linear sub-packetization, Index coding, Multi-Access Cache-aided Content Delivery Network

## I. INTRODUCTION

IT has been predicted in [1] that 82 percent of the global IT traffic will be video traffic by the year 2022. In [1], the authors also report that peak hour traffic is growing at a much faster rate than average internet traffic and the main contributor to this accelerated peak hour traffic is video content which tends to have a "prime time". Caching has been proposed as a method to shift some of the peak hour traffic to off-peak hours by placing contents into caches across the network during off-peak hours. Since most of the video content is generated well ahead of transmission, this type of traffic sits well within the caching framework.

In a seminal work [2], Maddah-Ali and Niesen introduced the notion of coding caching which jointly optimizes cache placement and content delivery by creating multi-cast opportunities and using coded transmissions to simultaneously deliver distinct contents to different users. The set-up in [2], which in this paper is called an $(N, K)$ centralized coded caching system, has a central server, which possesses $N$ files, coordinating cache placement as well as transmissions to $K$ different users, each possessing a dedicated cache memory which has a storage size of $M < N$ files. Since [2], researchers have explored several variants of the coded caching problem, including schemes with coded placement policies [3], [4], decentralized coded caching [5], device to device coded caching [6], online caching [7], caching with non-uniform file popularity and demands [8], [9], multi-access coded caching [10], [11], etc.

The placement phase of most of the coded caching schemes including [2] involve splitting each file into $F$ sub-packets, where, $F$ increases exponentially with the number of users $K$. This exponentially increasing sub-packetization creates two problems: the number of bits needed to index the sub-packets and the need for large file sizes. The first paper to look into the problem of sub-packetization was [12] and since then, an array of papers [12]-[18] have come up with coded caching schemes with reduced sub-packetization, a summary of which is given in [18]. Our paper also tries to address the sub-packetization problem by introducing a new coded caching scheme with linear sub-packetization inspired by the symmetric neighboring consecutive side information (SNCS) single unicast index coding problem introduced in [19]. While the previous works which proposed schemes with sub-exponential or linear sub-packetization required the number of users to be either extremely large [16], [18] or take values in some specific restricted sets [13], [18], our scheme does not impose any restriction on the number of users or the file size.

Further, for an $(N, K, L)$-CCDN (Cache-aided Content Delivery Network) which, as defined in [11], is a multi-access coded caching system with a single server having access to $N$ files, $K$ users and $K$ caches, $N \geq K$ such that each user has access to $L$ consecutive caches, of storage size $M$ files, with a cyclic wrap-around, we show that our delivery scheme can be used as a transmission policy which will satisfy the demands of all users. The paper [11] considers an $(N, K, L)$-CCDN at the memory points $M = \frac{iN}{K}$, $i \in \left[\left\lceil \frac{K}{L} \right\rceil\right] \cup \{0\}$ and for the $L \geq \frac{K}{2}$ regime, gives an upper and a lower bound on the achievable rate-memory trade-off (Corollary 2 and Theorem 3 in [11] respectively) and exact optimality results for some special cases (Theorem 5).

The main technical contributions in this paper are listed below.

- For a centralized coded caching system with $K$ users and $N$ files, a cache-placement policy where the number of sub-packets grows linearly with $K$, irrespective of file size, is introduced.
- For this placement scheme, a delivery policy which is instantaneously decodable (i.e, sub-packets involved in a transmission can be decoded by the users requesting them by utilizing only their cache contents and not any other transmission) is given.

- The rate-memory trade-off achieved by the delivery scheme is characterized.
- We show that the delivery scheme introduced in this paper can be used in the delivery phase of a multi-access coded caching setting considered in [11], which was called an $(N, K, L)$-CCDN in [11].
- Based on the rate-memory trade off achieved by our delivery scheme, we give a new outer bound for the rate-memory trade-off of an $(N, K, L)$-CCDN with $L \geq \frac{K}{2}$.
- For the special cases at which the exact rate-memory was characterized in [11], we show that our delivery scheme either achieves the optimal rate or if doesn't, the gap from optimality goes to zero with increasing number of users.

The rest of this paper is organized as follows. After stating the main results in this paper in section II, the new coded caching scheme is described in section III. Then, after a brief review of the multi-access coded caching setting considered in [11], we describe how the delivery scheme in section III can be used in the setting in [11] and a few results in that direction in section IV. Finally, the paper is concluded in section V.

*Notations*: For a prime power $q$, $\mathbb{F}_q$ denotes the finite field with $q$ elements. For a positive integer $n$, $[n]$ denotes the set $\{1, 2, \cdots, n\}$. The set of positive integers is denoted by $\mathbb{Z}^+$. A $t$-subset of $[n]$ is a subset of $[n]$ of size $t$. The symbol $\oplus$ denotes the XOR of its operands. Also, $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ and $\binom{n}{k} = 0$, when $n < 1$ or $n < k$. For an ordered set $\mathcal{S}$ with $n$ elements, the notation $\mathcal{S}(j)$ is used to denote the $j^{th}$ element in the set $\mathcal{S}$, for $j \in \{1, 2, \cdots, n\}$.

## II. MAIN RESULTS

**Theorem 1.** *For an $(N, K)$-coded caching system with cache memory size $M = \frac{iN}{K}$ for $i \in \{1, 2, \cdots, K\}$, the rate*

$$R\left(\frac{iN}{K}\right) = \left\lceil \frac{K(K-i)}{2 + \left\lfloor \frac{i}{K-i+1} \right\rfloor + \left\lfloor \frac{i-1}{K-i+1} \right\rfloor} \right\rceil \cdot \frac{1}{K}, \quad (1)$$

*is achievable with a linear sub-packetization $K$.*

*Proof.* This rate-memory pair can be achieved with a linear sub-packetization by using the placement policy described in subsection III-A followed by the delivery scheme given in subsection III-B. $\square$

**Theorem 2.** *Consider an $(N, K, L)$-CCDN with $L \geq \frac{K}{2}$. For cache memory size $M = \frac{N}{K}$, a new and improved upper bound for the rate $R(M)$ is given as*

$$R_{UB}(M) = \begin{cases} K - \left(K - \left\lceil \frac{K(K-L)}{2 + \left\lfloor \frac{L}{K-L+1} \right\rfloor + \left\lfloor \frac{L-1}{K-L+1} \right\rfloor} \right\rceil \frac{1}{K}\right) \frac{MK}{N} \\ \qquad \text{if } 0 \leq M \leq \frac{N}{K} \\ \left(\left\lceil \frac{K(K-L)}{2 + \left\lfloor \frac{L}{K-L+1} \right\rfloor + \left\lfloor \frac{L-1}{K-L+1} \right\rfloor} \right\rceil \frac{1}{K}\right) \left(2 - \frac{MK}{N}\right) \\ \qquad \text{if } \frac{N}{K} \leq M \leq \frac{2N}{K} \\ 0 \qquad \text{if } M \geq \frac{2N}{K}. \end{cases}$$
$$(2)$$

*Proof.* The $R_{UB}\left(\frac{N}{K}\right)$ is achieved by the delivery scheme in subsection III-B as explained in section IV-A. At $M = 0$, worst-case rate is $K$ and for $L \geq \frac{K}{2}$, the rate at $M = \frac{2N}{K}$ is

zero. The convex envelope of these three points is $R_{UB}(M)$. $\square$

## III. PROPOSED CODED CACHING SCHEMES

In this section, we describe the proposed coded caching scheme with a linear sub-packetization. Consider an $(N, K)$ centralized coded caching system with $K$ users, $\mathcal{U} = \{1, 2, \cdots, K\}$ and $N$ files, $\mathcal{W} = \{W_1, W_2, \cdots, W_N\}$. Each user has a dedicated cache which has a storage capacity of $M < N$ files. The cache content at user $k$ is denoted as $\mathcal{Z}_k$ and the set $\mathcal{Z} = \{\mathcal{Z}_1, \mathcal{Z}_2, \cdots, \mathcal{Z}_K\}$ denotes the overall cache contents. The scheme is described for those cases where cache fraction, $\frac{M}{N}$, takes values in $\{\frac{1}{K}, \frac{2}{K}, \cdots, 1\}$. We now describe the placement phase followed by the delivery scheme.

### A. Placement Phase:

Divide each file into $K$ sub-packets of equal size. The sub-packets corresponding to file $W_n$ are $\{W_{n,1}, W_{n,2}, \cdots, W_{n,K}\}$. Corresponding to a cache fraction of $\frac{M}{N} = \frac{i}{K}$, $i \in [K]$, the cache placement policy is as follows. The sub-packets $W_{n,k}, W_{n,k+1}, \cdots, W_{n,k+i-1}$ for all files $W_n \in \mathcal{W}$ are placed in the $k^{th}$ user's cache, i.e., for $k \in [K]$,

$$\mathcal{Z}_k = \{W_{n,k}, W_{n,k+1}, \cdots, W_{n,k+i-1} | \forall W_n \in \mathcal{W}\}$$

. Thus, each user has $i$ consecutive sub-packets, each of size $\frac{1}{K}$ units, of every file, in its cache. The memory occupied by these packets is $N \cdot \frac{i}{K}$ which is equal to $M$, thus satisfying the memory size constraint.

### B. Delivery Scheme

Consider an $(N, K)$ centralized coded caching system with user cache memory $M = \frac{iN}{K}$, $i \in [K]$. Following the cache placement in subsection III-A above, let the receiver $i$ demand the file $W_{d_i}$ and let $\mathbf{d} = \{d_1, d_2, \cdots, d_K\}$ denote the demand vector. Each receiver knows $i$ sub-packets of its demanded file and needs the remaining $(K - i)$ sub-packets. Thus, total number of demanded sub-packets is $K \cdot (K - i)$. For the demand vector $\mathbf{d}$, there exists a single unicast index coding problem [20], [21], [22] with number of users being equal to the number of demanded sub-packets where each user demands a distinct sub-file. Corresponding to each of the $K$ users in the coded caching set-up, there are $K - i$ receivers, each demanding a single sub-packet and all having the user's cache contents as side information, in the index coding problem. We give a delivery scheme which is an instantaneously decodable solution of the above index coding problem, i.e., we assume that the sub-packets involved in a particular transmission can be decoded, by the users requesting them, instantly upon receiving that transmission, using their cache contents only and not any other transmission.

Form the ordered set of demanded sub-packets $\mathcal{L} = \{W_{d_1,i}, W_{d_1,i+1}, \cdots, W_{d_1,K}, W_{d_2,i+1}, W_{d_2,i+2}, \cdots, W_{d_2,1}, \cdots, W_{d_k,i+k-1}, W_{d_k,i+k-2}, \cdots, W_{d_k,k-1}, \cdots, W_{d_K,i}, W_{d_K,i+1}, \cdots, W_{d_K,K-1}\}$. An algorithm which takes $\mathcal{L}$, $K$ and $i$ as inputs and generate the codewords for transmission are given in Algorithm 1 and the subroutines called in Algorithm

1 are given in the Algorithms 2, 3, 4 and 5. In the notation $W_{d_{u+a},p+a}$ used in the algorithms, the addition in the user index $u$ as well as the addition in the sub-packet index $p$ is performed with wrap-around w.r.t $K$. An explanation of the procedure is given below.

---

**Algorithm 1** Algorithm for generating the transmissions

---

**Require:** $\mathcal{L}$ and $K$, $i$.
1: Define $\Gamma \triangleq K - i + 1$.
2: Define $t \triangleq 2 + \left\lfloor \frac{i}{\Gamma} \right\rfloor + \left\lfloor \frac{i-1}{\Gamma} \right\rfloor$.
3: Generate the ordered set

$$T_0 = \{ W_{d_1,i+1}, W_{d_2,1}, W_{d_{1+\Gamma},i+1+\Gamma}, W_{d_{2+\Gamma},1+\Gamma}, \cdots ,$$
$$\cdots , W_{d_{1+\left\lfloor \frac{i}{\Gamma} \right\rfloor \Gamma},i+1+\left\lfloor \frac{i}{\Gamma} \right\rfloor \Gamma}, W_{d_{2+\left\lfloor \frac{i-1}{\Gamma} \right\rfloor \Gamma},1+\left\lfloor \frac{i-1}{\Gamma} \right\rfloor \Gamma} \},$$

with the $j^{\text{th}}$ element in $T_0$, i.e., $T_0(j)$ denoted as $W_{d_{u_j},p_j}$, where, $u_j$ is the user index and $p_j$ is the sub-packet index.
4: **if** $t$ is odd AND ($i < K - 2$) **then**
5: $\quad T_0(t) = W_{d_{u_t},p_t}$ is replaced by $W_{d_{u_t},p_t+1}$
6: **end if**
7: $\mathcal{C} = \emptyset$.
8: **while** $|\mathcal{C}| \leq \left\lceil \frac{K(K-i)}{t} \right\rceil$ **do**
9: $\quad T_1 = \emptyset$, flag = 0
10: $\quad$ **for** each $W_{d_{u_j},p_j} \in T_0$ **do**
11: $\quad\quad$ **if** $W_{d_{u_j},p_j} \in \mathcal{L}$ **then**
12: $\quad\quad\quad T_1 = T_1 \cup W_{d_{u_j},p_j}$.
13: $\quad\quad$ **else**
14: $\quad\quad\quad$ **if** $T_1 == \emptyset$ AND $|\mathcal{L}| == K$ **then**
15: $\quad\quad\quad\quad T_1 = \text{SubRoutine}(\mathcal{L}, K, i)$
16: $\quad\quad\quad$ **else**
17: $\quad\quad\quad\quad [T_1, \text{flag}] = \text{Update}(W_{d_{u_j},p_j}, \mathcal{L}, K, T_1, \text{flag})$
18: $\quad\quad\quad$ **end if**
19: $\quad\quad$ **end if**
20: $\quad$ **end for**
21: $\quad \mathcal{L} = \mathcal{L} \setminus T_1$
22: $\quad T_0 = \emptyset$.
23: $\quad c = 0$
24: $\quad$ **for** each $W_{d_{u_j},p_j} \in T_1$ **do**
25: $\quad\quad c = c \bigoplus W_{d_{u_j},p_j}$
26: $\quad\quad T_0 = T_0 \cup W_{d_{u_j+1},p_j+1}$
27: $\quad$ **end for**
28: **end while**
29: **return** Code $\mathcal{C}$ for transmission.

---

The first codeword is formed as

$$c = \sum_\beta \underbrace{W_{d_{1+\beta\Gamma},i+1+\beta\Gamma}}_{\text{term1}} + \underbrace{W_{d_{2+\beta\Gamma},1+\beta\Gamma}}_{\text{term2}}, \quad (3)$$

where, $\beta$ takes the values $0, 1, \cdots \left\lfloor \frac{i}{\Gamma} \right\rfloor$ for term 1 and the values $0, 1, \cdots \left\lfloor \frac{i-1}{\Gamma} \right\rfloor$ for term 2. Thus, the codeword $c$ is a sum of $t = 2 + \left\lfloor \frac{i}{\Gamma} \right\rfloor + \left\lfloor \frac{i-1}{\Gamma} \right\rfloor$ sub-packets.

After generating a codeword, the corresponding sub-packets are removed from the set $\mathcal{L}$, and the next codeword is generated by incrementing both the file index and the sub-packet index by one for all the sub-packets involved in the current code-word. If a term not present in $\mathcal{L}$ appears, it is replaced by another sub-packet, the rules for choosing which

---

are given in the Rule() subroutine in Algorithm 4. In some special cases, at the end of an iteration, when only $K$ sub-packets remain in $\mathcal{L}$, a special construction of the set $T_1$ is needed which is given by the subroutine in Algorithm 2. This is especially the case when $i \leq \frac{K}{2}$ and $K - i$ is odd.

Since, each codeword is the sum of $t$ elements and there are $K(K - i)$ sub-packets in $\mathcal{L}$, the number of transmissions required is $\Lambda = \left\lceil \frac{K(K-i)}{t} \right\rceil$. Hence, the codeword generation is continued for $\Lambda$ iterations. Each of the generated codeword is of size $\frac{1}{K}$ units and hence, the expression for the rate of transmission $R(M)$ can be characterized as

$$R_{\text{New}}\left( \frac{iN}{K} \right) = \frac{\Lambda}{K} = \left\lceil \frac{K(K-i)}{2 + \left\lfloor \frac{i}{K-i+1} \right\rfloor + \left\lfloor \frac{i-1}{K-i+1} \right\rfloor} \right\rceil \cdot \frac{1}{K}. \quad (4)$$

---

**Algorithm 2** SubRoutine($\mathcal{L}, K, i$)

---

**Require:** $\mathcal{L}$ and $K$, $i$.
1: $\Gamma = K - i + 1$.
2: $t = 2 + \left\lfloor \frac{i}{\Gamma} \right\rfloor + \left\lfloor \frac{i-1}{\Gamma} \right\rfloor$.
3: $T = W_{d_1,k}$, for some $k \in [K]$ such that $W_{d_1,k} \in \mathcal{L}$.
4: **for** $j = 1$ to $t - 1$ **do**
5: $\quad T = T \cup W_{d_{1+\left\lfloor \frac{jK}{t} \right\rfloor},k+\left\lfloor \frac{jK}{t} \right\rfloor}$.
6: **end for**
7: **return** $T$.

---

**Algorithm 3** Update($W_{d_{u_j},p_j}, \mathcal{L}, \mathcal{Z}, T_1$, flag)

---

1: **if** flag == 0 **then**
2: $\quad$ **for** k = 1:4 **do**
3: $\quad\quad \hat{x} = \text{Rule}(W_{d_{u_j},p_j}, k)$
4: $\quad\quad$ **if** Check($\hat{x}, \mathcal{Z}, \mathcal{L}, T_1$) **then**
5: $\quad\quad\quad T_1 = T_1 \cup \hat{x}$
6: $\quad\quad\quad$ flag = k
7: $\quad\quad\quad$ **break**
8: $\quad\quad$ **end if**
9: $\quad$ **end for**
10: **else if** flag == 1 OR flag == 3 **then**
11: $\quad$ flag = flag + 1
12: $\quad T_1 = T_1 \cup \text{Rule}(W_{d_{u_j},p_j}, \text{flag})$
13: **else if** flag == 2 OR flag == 4 **then**
14: $\quad$ flag = flag - 1
15: $\quad T_1 = T_1 \cup \text{Rule}(W_{d_{u_j},p_j}, \text{flag})$
16: **end if**
17: **return** $T_1$, flag

---

**Remark 1.** *When $i = 1$ or $i = K - 1$, for any $K$ and $N$, the placement is same as that in Maddah-Ali-Niesen scheme [2] and the delivery scheme in Algorithm 1 gives the same set of transmissions as that in [2]. Hence, the rate achieved at these points, i.e., $R\left( \frac{N}{K} \right) = \frac{K-1}{2}$ and $R\left( \frac{(K-1)N}{K} \right) = \frac{1}{K}$, are the same as that achieved by Maddah-Ali-Niesen scheme and hence optimal under the constraint of uncoded placement.*

**Algorithm 4** Rule($W_{d_{u_j},p_j}$, flag)

---

1: **if** flag == 1 **then**
2:    $\hat{x} = W_{d_{u_j},p_j+1}$.
3: **else if** flag == 2 **then**
4:    $\hat{x} = W_{d_{u_j+1},p_j}$.
5: **else if** flag == 3 **then**
6:    $\hat{x} = W_{d_{u_j-1},p_j}$.
7: **else if** flag == 4 **then**
8:    $\hat{x} = W_{d_{u_j},p_j-1}$.
9: **end if**
10: **return** $\hat{x}$.

---

**Algorithm 5** Check($W_{d_{u_j},p_j}, \mathcal{Z}, \mathcal{L}, T$)

---

1: **if** $W_{d_{u_j},p_j} \notin \mathcal{L}$ **then**
2:    **return** 0
3: **else**
4:    **for** each $W_{d_{u_k},p_k} \in T$ **do**
5:      **if** $p_j \notin Z_{u_k}$ OR $p_k \notin Z_{u_j}$ **then**
6:        **return** 0
7:      **end if**
8:    **end for**
9: **end if**
10: **return** 1.

---

**Remark 2.** *When $1 < i \le \frac{K}{2}$ , for any $N$ and $K$, an instantaneously decodable transmission can only contain the combination of two sub-packets at the most. The code given by Algorithm 1 is given by*

$$W_{d_{1+a},i+a+k} + W_{d_{1+k+a},1+a}, \tag{5}$$

*for $a \in \{0, 1, ...K-1\}$, $k \in \{1, 2, \lfloor \frac{K-i}{2} \rfloor\}$.*

*The above equation gives $K \cdot \lfloor \frac{K-i}{2} \rfloor$ transmissions. When $(K - i)$ is odd, we also need to make the following set of transmissions.*

$$W_{d_{1+a},i+\lceil \frac{K-i}{2} \rceil+a} + W_{d_{\lfloor \frac{K}{2} \rfloor+1+a},\lceil \frac{i}{2} \rceil+a}, \tag{6}$$

*for $a \in \{0, 1, \cdots, \lceil \frac{K}{2} \rceil\}$.*

*The equation (6) above results in an additional $\lceil \frac{K}{2} \rceil$ transmissions when $(K - i)$ is odd. Thus, for both $(K - i)$ odd as well as even, there is a total of $\lceil \frac{K(K-i)}{2} \rceil$ transmissions, each of size $\frac{1}{K}$, giving the rate $R(\frac{iN}{K}) = \lceil \frac{K(K-i)}{2} \rceil \cdot \frac{1}{K}$.*

**Decodability** : Decodability of the above delivery scheme is explained by considering three cases as follows.
*Case I : $i = 1$ or $i = K - 1$* : For this case, as explained in Remark 1, the transmissions given by Algorithm 1 is a reordered form of the Maddah-Ali-Niesen scheme in [2] and hence decodability is assured.

*Case II : $1 < i \le \frac{K}{2}$* : For this case, consider the transmissions given by (5). In a single transmission, only two users, $u_1$ and $u_2$, with user indices $1 + a$ and $1 + k + a$ respectively, where $a$ and $k$ takes values as given in (5) are involved. With respect to these users, the transmission in (5) can be re-written as $W_{d_{u_1},u_2+i-1} + W_{d_{u_2},u_1}$. Since the user

$u_j$ knows $i$ consecutive sub-packets with indices from $u_j$ to $u_j + i - 1$, both the users involved in a transmission know the sub-packet not requested by it. Now, consider the transmission given in (6). Here, $u_1 = 1 + a$ and $u_2 = \lfloor \frac{K}{2} \rfloor + 1 + a$. The user $u_1$ knows sub-packets with indices from $1 + a$ to $i + a$ of all files, it knows $W_{d_{u_2},\lceil \frac{i}{2} \rceil+a}$. Similarly, $u_2$ knows the sub-packets $\lfloor \frac{K}{2} \rfloor + 1 + a$ to $\lfloor \frac{K}{2} \rfloor + i + a$ and hence knows $W_{d_{u_1},\lceil \frac{K-i}{2} \rceil+i+a}$.

*Case III : $i > \frac{K}{2}$* : Consider the first codeword $c$ given in (3). The terms involved in this codeword can be grouped into two sets as follows:

TABLE I: Table containing terms in (3).

| Column 1 | Column 2 |
|---|---|
| $W_{d_1,i+1}$ | $W_{d_2,1}$ |
| $W_{d_{1+\Gamma},2}$ | $W_{d_{2+\Gamma},1+\Gamma}$ |
| $W_{d_{1+2\Gamma},2+\Gamma}$ | $W_{d_{2+2\Gamma},1+2\Gamma}$ |
| $W_{d_{1+3\Gamma},2+2\Gamma}$ | $W_{d_{2+3\Gamma},1+3\Gamma}$ |
| $\vdots$ | $\vdots$ |
| $W_{d_{1+\lfloor \frac{i}{\Gamma} \rfloor\Gamma},2+(\lfloor \frac{i}{\Gamma} \rfloor-1)\Gamma}$ | $W_{d_{2+\lfloor \frac{i-1}{\Gamma} \rfloor\Gamma},1+\lfloor \frac{i-1}{\Gamma} \rfloor\Gamma}$ |

A term in $c$ is of the form $W_{d_{u_j},p_j}$, where, $u_j$ is the user index and $p_j$ is the sub-packet index. If $\lfloor \frac{i}{\Gamma} \rfloor > \lfloor \frac{i-1}{\Gamma} \rfloor$, there are odd number of terms in $c$ and the last term is $W_{d_{1+\lfloor \frac{i}{\Gamma} \rfloor\Gamma},3+(\lfloor \frac{i}{\Gamma} \rfloor-1)\Gamma}$ if $K - i > 2$ and $W_{d_{1+\lfloor \frac{i}{\Gamma} \rfloor\Gamma},2+(\lfloor \frac{i}{\Gamma} \rfloor-1)\Gamma}$ if $K - i \le 2$. If $\lfloor \frac{i}{\Gamma} \rfloor = \lfloor \frac{i-1}{\Gamma} \rfloor$, then there are even number of terms in $c$ and the last term is $W_{d_{2+\lfloor \frac{i-1}{\Gamma} \rfloor\Gamma},1+\lfloor \frac{i-1}{\Gamma} \rfloor\Gamma}$. In all these cases, the largest sub-packet index that appears in $c$ is $i + 1$ corresponding to the first term $W_{d_1,i+1}$ and it can be verified that the largest user index that can appear in $c$ is also $i + 1$. Similarly the smallest user-index as well as sub-packet index in $c$ is 1. Hence $1 \le u_j, p_j \le i + 1$, $j \in \{1, 2, \cdots, t\}$. Because of this, it can be seen that the users in the first row, i.e, users 1 and 2, knows all the sub-packets except their own demanded sub-packet. Further, note that the user indices appear in $c$ in an increasing order $\{1, 2, 1 + \Gamma, 2 + \Gamma, \cdots, 1 + \lceil \frac{i}{\Gamma} \rceil\}$. Similarly, the sub-packet indices except the first one, i.e., $i + 1$, appears in an increasing order $\{1, 2, 1 + \Gamma, \cdots, \}$

Now consider a general user in Column 1 of Table I, which is of the form $u_k = 1 + k\Gamma$. This user knows all sub-packet indices from $u_k$ to $u_k+i-1$, i.e., from $1+k\Gamma$ to $1+k\Gamma+i-1$, which can be computed as $u_k + i - 1 = 1 + k\Gamma + i - 1 = k(K - i + 1) + i = (k - 1)\Gamma + K - i + 1 + i = (k-1)\Gamma + 1$, considering wrap-around w.r.t $K$. Since, the sub-packet indices appear in an increasing order, to show that user $u_k$ knows all the sub-packets in $c$, except its own demanded sub-packet, it suffices to show that it knows the sub-packets of its preceding and succeeding terms. It can be seen from Table I that the index of the sub-packet demanded by the user succeeding the user $u_k$, (which is the user with index $2+k\Gamma$), is $1+k\Gamma = u_k$ which is known to user $u_k$. Similarly, the preceding user index is $2 + (k-1)\Gamma$ and its demanded sub-packet is $1 + (k-1)\Gamma = u_k + i - 1$. Hence, a general user in Column 1 can decode

its required sub-packet from $c$ as it knows all the other sub-packets involved.

Now, consider a general user $u_k$ in Column 2 of Table I, which is of the form $u_k = 2 + k\Gamma$. This user knows sub-packets with indices $u_k$ to $u_k + i - 1 = 2 + (k-1)\Gamma$. The sub-packet demanded by the preceding and succeeding users of $u_k$ in $c$ are of indices $2 + (k-1)\Gamma = u_k + i - 1$ and $2 + k\Gamma = u_k$. Hence, similar to a general user in Column 1, a general user $u_k$ in Column 2 also knows all the sub-packets involved in $c$, except its own demanded sub-packet. Hence, we saw each of the $t$ users whose demanded sub-packets appear in a codeword $c$ knows all $(t-1)$ other sub-packets involved in $c$ and hence can decode their respective demanded sub-packet from the codeword using their cache contents. Since the other code-words are obtained by incrementing the user and sub-packet indices of all the terms in $c$ by equal amounts, the users involved in all the transmissions know all but its own demanded sub-packet.

When a term already removed from $\mathcal{L}$ in a previous iteration appears again, the Update() subroutine gives another sub-packet present in $\mathcal{L}$ by using one out of four rules. While doing this, the Check() subroutine is invoked to verify that the updated sub-packet is present in the caches of all the other users involved in that transmission and vice-versa. Hence, the code obtained from Algorithm 1 is instantaneously decodable.

### C. Illustrating Example

To illustrate the placement and delivery schemes described above, we give the following example.

**Example 1.** *Consider an $(N = 6, \ K = 6)$-centralized coded caching system with $M = 4$. The cache contents and for the demand vector, $\mathbf{d} = \{1, 2, \cdots, 6\}$, the sub-packets requested by each user is shown in Table II.*

TABLE II: Example 1 - Cache contents and sub-packets required at each user for the demand vector $\mathbf{d} = \{1, 2, 3, 4, 5, 6\}$.

| User,$i$ | Cache contents, $\mathcal{Z}_i$ | Reqd. sub-packets |
|---|---|---|
| 1 | $\{W_{n,1}, W_{n,2}, W_{n,3}, W_{n,4} \ \forall n \in [6]\}$ | $\{W_{1,5}, W_{1,6}\}$ |
| 2 | $\{W_{n,2}, W_{n,3}, W_{n,4}, W_{n,5} \ \forall n \in [6]\}$ | $\{W_{2,6}, W_{2,1}\}$ |
| 3 | $\{W_{n,3}, W_{n,4}, W_{n,5}, W_{n,6} \ \forall n \in [6]\}$ | $\{W_{3,1}, W_{3,2}\}$ |
| 4 | $\{W_{n,4}, W_{n,5}, W_{n,6}, W_{n,1} \ \forall n \in [6]\}$ | $\{W_{4,2}, W_{4,3}\}$ |
| 5 | $\{W_{n,5}, W_{n,6}, W_{n,1}, W_{n,2} \ \forall n \in [6]\}$ | $\{W_{5,3}, W_{5,4}\}$ |
| 6 | $\{W_{n,6}, W_{n,1}, W_{n,2}, W_{n,3} \ \forall n \in [6]\}$ | $\{W_{6,4}, W_{6,5}\}$ |

*Delivery Scheme : The transmissions obtained from the scheme described in subsection III-B are given below.*

$$W_{5,4} + W_{1,5} + W_{2,1} + W_{4,2}$$
$$W_{6,5} + W_{2,6} + W_{3,2} + W_{5,3}$$
$$W_{1,6} + W_{3,1} + W_{4,3} + W_{6,2}$$

*The rate achieved using this scheme is $R(2) = \frac{3}{6} = \frac{1}{2}$ with a sub-packetization of $K = 6$. For the same set-up, Maddah-Ali-Niesen scheme [2] attains a rate of $R_{M\text{-}N} = \frac{2}{5}$ with a sub-packetization of $\binom{6}{4} = 15$.*

## IV. APPLICATION OF THE DELIVERY SCHEMES FOR A MULTI-ACCESS CODED CACHING SYSTEM

We now explain how the delivery scheme described in section III can be employed as a transmission scheme for an $(N, K, L)$-CCDN and how the rate achieved by our scheme compares against the upper and lower bounds and optimality characterizations in [11].

### A. Delivery Scheme for an $(N, K, L)$-CCDN

For an $(N, K, L)$-CCDN, $L \in [K]$, with cache memory size $M = \frac{iN}{K}, i \in \{0, 1, \cdots, \lceil \frac{K}{L} \rceil\}$, each user has access to $L$ consecutive caches and thus a total cache memory of $\frac{iLN}{K}$. This is similar to a centralized coded caching set-up with each user having a dedicated cache memory of $\frac{iLN}{K}$ except for the total amount of cache memory available in both the systems. In the $(N, K, L)$-CCDN, the total memory available is $K \cdot \frac{M}{N} = iN$, whereas, in the (N,K)-coded caching system, with $M = \frac{iLN}{K}$, it is $iLN$. Now, let us see how the delivery policy developed in section III for an $(N, K)$ centralized coded caching system can be used as a delivery scheme for an $(N, K, L)$-CCDN.

*1) Case I - $L \geq \frac{K}{2}$:* Let us first consider $L \geq \frac{K}{2}$. For this case, $\lceil \frac{K}{L} \rceil = 2$ and hence $M$ only takes values in $\{0, \frac{N}{K}, \frac{2N}{K}\}$ out of which the only non-trivial memory point is $M = \frac{N}{K}$ as at $M = 0$, there is no cache memory and the worst-case rate is $K$, whereas $R = 0$ at $M = \frac{2N}{K}$. For $M = \frac{N}{K}$, the placement scheme in [11] specializes as follows. Each file is split into $K$ equal parts and the $i^{\text{th}}$ sub-file is stored in the $i^{\text{th}}$ cache and user $k$ has access to the caches $k, k+1, \cdots, k+L$, which implies it has access to the sub-files indexed by $k, k+1, \cdots, k+L$ of all files. This is exactly the same as the contents of the $k^{\text{th}}$ user's cache following the placement given in subsection III-A.

Now, for a demand vector $\mathbf{d} = \{d_1, d_2, \cdots, d_K\}$, in both the settings, each user needs $K - L$ sub-packets of its demanded file. Thus, the index coding problem arising in both the settings are the same and hence admits the same solution. Thus, the delivery scheme in section III is a valid solution for the multi-access coded caching setting in an $(N, K, L)$-CCDN with $M = \frac{N}{K}$ and $L \geq \frac{K}{2}$ as well.

*2) Case II - $L < \frac{K}{2}$:* For this case, we need to consider the memory points $M = \frac{iN}{K}, i \in \{0, 1, \cdots, \lceil \frac{K}{L} \rceil\}$ out of which worst-case rate $R = K$ at $M = 0$ and $R = 0$ at $M = \lceil \frac{K}{L} \rceil \cdot \frac{N}{K}$. At the remaining memory points, the placement scheme in [11] splits each file into $F(i, L) = \binom{K-iL+i-1}{i-1} \cdot \frac{K}{i}$ subfiles. $F(i, L) = K$ only for $i = 1$ and $i = \lfloor \frac{K}{L} \rfloor < \frac{K}{L}$. For these two cases, placement in [11] is same as our placement scheme as explained in Case I above. At all other values of $i$, the value of $F(i, L)$ is strictly greater than $K$. So, for a cache memory of $M = \frac{N}{K}$ as well as $M = \lfloor \frac{K}{L} \rfloor \frac{N}{K}$ in an $(N, K, L)$-CCDN with $L < K$, following the placement phase, corresponding to a demand vector, $\mathbf{d} = \{d_1, d_2, \cdots, d_K\}$, the index coding problem seen is the same as that in an $(N, K)$- coded caching network with cache memory $LM$ and hence our delivery scheme can be used for transmission.
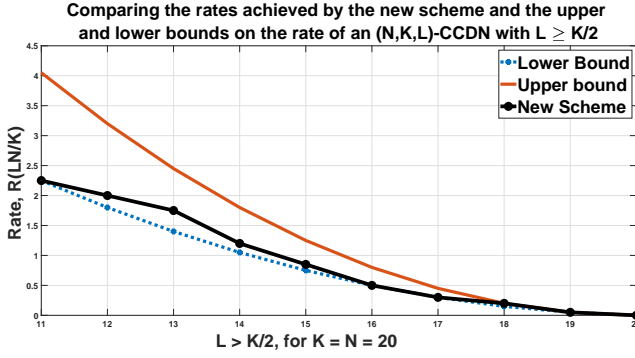
Fig. 1: Rate-Memory trade-off Comparison

TABLE III: Comparison between the optimal rates (in Theorem 5, [11]) and that achieved by the new scheme

| $L$ | $R^* \left( \frac{N}{K} \right)$ | $R_{\text{New}} \left( \frac{N}{K} \right)$ |
|---|---|---|
| $K-1$ | $\frac{1}{K}$ | $\frac{1}{K}$ |
| $K-2$ | $\frac{3}{K}$ | $\begin{cases} \frac{3}{K} & \text{if } K=3n,\ n \in \{2,3,\cdots,..\} \\ \frac{4}{K} & \text{otherwise} \end{cases}$ |
| K-3 | $\frac{6}{K}$ | $\begin{cases} \frac{9}{K} & \text{if } K=6 \\ \frac{8}{K} & \text{if } K=5,10 \\ \frac{6}{K} & \text{if } K=4n,\ n \in \mathbb{Z}^+ \\ \frac{7}{K} & \text{otherwise} \end{cases}$ |
| $K - \frac{K}{s} + 1$ $s \in \mathbb{Z}^+$ | $\frac{(K-s)}{2s^2}$ | $\frac{(K-s)}{2s^2}$ |

### B. New Improved Upper Bound for $L \geq \frac{K}{2}$

A comparison plot showing the existing upper bound as well as lower bound and the rate achieved by the new scheme for the case $N = K = 10$ and $L > \frac{K}{2}$ is given in Fig. 1. Since our scheme performs better than the existing upper bound and since for $L < \frac{K}{2}$, the only bound on rate available is the upper bound (Theorem 1 in [11]), the rate achieved by the newly introduced scheme in section III can be used to derive an improved upper bound, which is given in Theorem 2. The improved upper bound based on our scheme is very close to the lower bound in general, and achieves it at some points.

### C. Exact Optimality Cases

In [11], for exact optimality characterizations (Theorem 5), only three memory points, $M \in \left\{0, \frac{N}{K}, \frac{2N}{K}\right\}$ are considered. Since the optimality results are given for the $L \geq \frac{K}{2}$ regime, among these three points, $M = \frac{N}{K}$ is the only non-trivial point as at $M = 0$, there is no cache memory and the worst-case rate is $K$, whereas $R = 0$ at $M = \frac{2N}{K}$. A comparison between the optimal rate and that achieved by our scheme for the cases in Theorem 5 of [11] is given in Table III. We can see that for all the cases covered, our scheme either achieves the optimal rate-memory trade-off or the gap from optimality goes to zero with increasing number of users.

### V. CONCLUDING REMARKS

In this paper, we looked at the problem of sub-packetization which poses a major challenge while implementing a coded

caching protocol. Towards this end, we developed a coded caching scheme, with a sub-packetization that varies linearly with the number of users, inspired by the one-sided SNCS index coding problem. We also showed that the delivery scheme introduced in this paper is a solution for the index coding problem arising in the multi-access coded caching set-up called an $(N, K, L)$-CCDN considered in [11] and that the rate-expression corresponding to the newly introduced delivery scheme can be used to derive a new upper-bound for the rate achieved in an $(N, K, L)$-CCDN. Further, for the special cases for which the optimal rate-memory trade-off was characterized in [11], we analyzed the performance of our delivery scheme.

### REFERENCES

[1] *Cisco Visual Networking Index: Forecasts and Trends, 2017-2022.* [Online]. Available : https://cyrekdigital.com/pl/blog/content-marketing-trendy-na-rok-2019/white-paper-c11-741490.pdf
[2] M. A. Maddah-Ali and U. Niesen, "Fundamental Limits of Caching," in *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856-2867, May 2014.
[3] Z. Chen, P. Fan, and K. B. Letaief, "Fundamental limits of caching: improved bounds for users with small buffers," *IET Communications*, vol. 10, no. 17, pp. 2315-2318, Nov. 2016.
[4] J. Gómez-Vilardebó, "Fundamental Limits of Caching: Improved Rate-Memory Tradeoff With Coded Prefetching," in *IEEE Transactions on Communications*, vol. 66, no. 10, pp. 4488-4497, Oct. 2018.
[5] M. A. Maddah-Ali and U. Niesen, "Decentralized Coded Caching Attains Order-Optimal Memory-Rate Tradeoff," in *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1029-1040, Aug. 2015
[6] M. Ji, G. Caire and A. F. Molisch, "Fundamental Limits of Caching in Wireless D2D Networks," in *IEEE Transactions on Information Theory*, vol. 62, no.2, pp. 849-869, Feb 2016.
[7] R. Pedarsani, M. A. Maddah-Ali and U. Niesen, "Online Coded Caching," in *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 836-845, April 2016.
[8] U. Niesen and M. A. Maddah-Ali, "Coded Caching With Nonuniform Demands," in *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 1146-1158, Feb. 2017
[9] J. Zhang, X. Lin and X. Wang, "Coded caching under arbitrary popularity distributions," in *Proc. Information Theory and Applications Workshop (ITA)*, San Diego, CA, USA, pp. 98-107, Feb 2015.
[10] J. Hachem, N. Karamchandani and S. N. Diggavi, "Coded Caching for Multi-level Popularity and Access," in *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 3108-3141, May 2017.
[11] K. S. Reddy and N. Karamchandani, "Rate-Memory Trade-off for Multi-Access Coded Caching With Uncoded Placement," in *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3261-3274, June 2020.
[12] Q. Yan, M. Cheng, X. Tang and Q. Chen, "On the Placement Delivery Array Design for Centralized Coded Caching Scheme," in *IEEE Transactions on Information Theory*, vol. 63, no. 9, pp. 5821-5833, Sept. 2017.
[13] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," in *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 57555766, Aug 2018.
[14] Q. Yan, X. Tang, Q. Chen, and M. Cheng, "Placement delivery array design through strong edge coloring of bipartite graphs," in *IEEE Communications Letters*, vol. 22, no. 2, pp. 236239, Feb 2018.
[15] L. Tang and A. Ramamoorthy, "Coded caching schemes with reduced subpacketization from linear block codes," in *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 30993120, April 2018.
[16] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using ruzsa-szemeredi graphs, in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, pp. 12371241, June 2017.

[17] M. Cheng, Q. Yan, X. Tang, and J. Jiang, "Coded caching schemes with low rate and subpacketizations," arXiv:1703.01548, Mar. 2017.

[18] H H S Chittoor, Prasad Krishnan, K V Sushena Sree and Bhavana MVN, "Subexponential and Linear Subpacketization Coded Caching via Line Graphs and Projective Geometry," arXiv:2001.00399, Jan. 2020.

[19] H. Maleki, V. R. Cadambe and S. A. Jafar, "Index CodingAn Interference Alignment Perspective," in *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5402-5432, Sept. 2014

[20] Y. Birk and T. Kol, "Informed-source coding-on-demand (ISCOD) over broadcast channels," in *Proc. IEEE INFOCOM '98*, San Francisco, CA, USA, vol.3, pp. 1257-1264, April 1998.

[21] Z. Bar-Yossef, Y. Birk, T. S. Jayram and T. Kol, "Index Coding With Side Information," in *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479-1494, March 2011.

[22] L. Ong, C. K. Ho and F. Lim, "The Single-Uniprior Index-Coding Problem: The Single-Sender Case and the Multi-Sender Extension," in *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3165-3182, June 2016.