An Embedded Index Code Construction Using Sub-packetization

Shanuja Sasi[1], Vaneet Aggarwal[2], and B. Sundar Rajan[1]
[1] Indian Institute of Science, Bengaluru [2] Purdue University, Indiana
E-mail: shanuja@iisc.ac.in, vaneet@purdue.edu, bsrajan@iisc.ac.in

2020 IEEE Information Theory Workshop (ITW) | 978-1-7281-5962-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/ITW46852.2021.9457585

***Abstract*—A variant of the index coding problem (ICP), the embedded index coding problem (EICP) was introduced in [A. Porter and M. Wootters, "Embedded Index Coding," ITW, Sweden, 2019] which was motivated by its application in distributed computing where every user can act as sender for other users and an algorithm for code construction was reported. The construction depends on the computation of min-rank of a matrix, which is computationally intensive. In [A.A. Mahesh, N. S. Karat and B. S. Rajan, "Min-rank of Embedded Index Coding Problems," ISIT, 2020], the authors have provided an explicit code construction for a class of EICP - *Consecutive and Symmetric Embedded Index Coding Problem (CS-EICP)*. We introduce the idea of sub-packetization of the messages in index coding problems to provide a novel code construction for CS-EICP in contrast to the scalar linear solutions provided in the prior works. For CS-EICP, the normalized rate, which is defined as the number of bits transmitted by all the users together normalized by the total number of bits of all the messages, for our construction is lesser than the normalized rate achieved by Mahesh *et al.*, for scalar linear codes.**

I. Introduction

Index coding problem (ICP) is a canonical problem in network information theory, that provides a simple yet rich model for several important engineering problems in network communication, such as content broadcasting, peer-to-peer communication, distributed caching, device-to-device relaying, distributed storage, and interference management [1]–[5]. The authors of [6] introduced a variant of ICP, called embedded index coding problem (EICP), where each node can be both sender and user at the same time. This problem is motivated by applications in distributed computation and distributed storage. It is a special case of multi-sender ICP [7]–[9], where the set of users and senders are the same. It has got application in vehicular ad-hoc networks (VANETs) which have gained popularity with their importance in intelligent transport systems [10]. In [11], scalar linear index coding techniques have been applied to reduce the number of transmissions required for data exchange during the Vehicle to Vehicle (V2V) communication phase which is an integral part of collaborative message dissemination in VANETs.

EICP consists of a set of users where each user already has a subset of messages and demands another subset of messages. Each user is fully aware of the content available at all other users and can communicate to all its peers through an error-free broadcast channel. The goal is to minimize the number of bits transmitted by all the users such that each user retrieves whatever they have demanded. There are no separate senders involved in this setting. Some results establishing relationships between single sender (centralized) index coding and EICP have been provided in [6]. In particular, it is shown that, the optimal code length for an EICP is only a factor of two worse than the optimal code length for a single sender index coding problem with the same setting. A heuristic algorithm has also been proposed for EICP. In [12], for EICP, a notion of side-information matrix was introduced. The length of an optimal scalar linear index code was derived to be equal to the min-rank of the side-information matrix.

In this paper, we consider a specific class of embedded index coding problem, defined as *Consecutive and Symmetric Embedded Index Coding Problem (CS-EICP)*. We assume that the cardinality of the side-information is same for all the users. The *normalized rate* is defined as the total number of bits transmitted by all the users together normalized by the total bits of all the messages.

In [6], the proposed heuristic algorithm for EICP involves calculating min-rank of a graph, by searching over all possible fitting matrices, which is computationally complex. In [12], the CS-EICP was studied as 'one-sided neighboring side-information problem'. The authors had characterized the length of the optimal scalar linear index code for CS-EICP to be $N-s+1$, where N represents the number of users as well as messages and s represents the cardinality of side-information available at each user. A scalar linear code achieving this length was also constructed. Hence the normalized rate is $\frac{N-s+1}{N}$. In this paper, we provide an explicit code construction for the CS-EICP by appropriately invoking sub-packetization of the messages. The normalized rate achieved in our scheme is $\frac{1}{\lceil \frac{s}{N-s} \rceil}$, if $s > \frac{N}{2}$ and $\frac{\lceil \frac{N-s}{s-1} \rceil}{1+\lceil \frac{N-s}{s-1} \rceil}$, if $s \leq \frac{N}{2}$. For certain ranges of values of s, we prove that it is less than $\frac{N-s+1}{N}$.

One of the special cases of EICP is when the users demand all the messages which are not in the side-information. This special case was studied as Cooperative Data Exchange (CDE) problem in [13], where there is a set of M messages and N users which demand the whole message set. Each user already has a subset of the messages available as side-information. Upper and lower bounds on the minimum number of transmissions are provided in [13]. For the case when all the users have the same number of messages, i.e. s, as side-information, the lower bound on the number of transmissions required is $M - s + 1$, i.e., the normalized rate is lower bounded by $\frac{M-s+1}{M}$. If our scheme is specialized to CDE problem, then the

978-1-7281-5962-1/21/$31.00 ©2021 IEEE

normalized rate achieved in our scheme is $\frac{1}{\lceil \frac{s}{N-s} \rceil}$, if $s > \frac{N}{2}$ and $\frac{\lceil \frac{N-s}{s-1} \rceil}{1+\lceil \frac{N-s}{s-1} \rceil}$, if $s \leq \frac{N}{2}$. Here also, for some cases, we prove that it is less than $\frac{M-s+1}{M}$.

A. Vector linear code and sub-packetization scheme.

An index coding scheme is said to be linear if the transmitted index code symbols are linear combinations of the messages. A scalar linear code uses only one instant of the M message symbols to obtain the index code symbols whereas a vector linear code uses multiple instants of M messages to obtain the index code symbols. For example, if the sender uses two instants of M messages and sends n linear index code symbols, then it means that n linear combinations of $2M$ messages are broadcast and the code is a vector linear code.

In sub-packetization scheme that we introduce in this paper for index coding problems, we do not use multiple instants of messages. We use only one instant of the M message symbols while we split each message of size d bits into z blocks. We assume that d is sufficiently large such that this splitting of message into z blocks of equal sizes is possible. The size of each block is $d_1 = \frac{d}{z}$ bits and each block is assumed to be from a finite field $\mathbb{F}_{2^{d_1}}$. The coded symbols transmitted are a linear combination of these blocks rather than the linear combination of the entire messages. Sub-packetization is extensively used and studied in the coded caching literature.

B. Our Contributions

The contributions of this paper is summarized as follows.

- We introduce the idea of sub-packetization in index coding problems to provide code construction for a special class of EICP, namely *Consecutive and Symmetric Embedded Index Coding Problem (CS-EICP)*.
- We show that, for CS-EICP, the normalized rate achieved in our scheme is $\frac{1}{\lceil \frac{s}{N-s} \rceil}$, if $s > \frac{N}{2}$ and $\frac{\lceil \frac{N-s}{s-1} \rceil}{1+\lceil \frac{N-s}{s-1} \rceil}$, if $s \leq \frac{N}{2}$. We prove that, when $(s-1)$ divides $(N-1)$ or $(N-s)$ divides $(N-1)$ or $s > \frac{2N+1-\sqrt{4N+1}}{2}$, this is less than the normalized rate $\frac{N-s+1}{N}$ achieved in [12] using scalar linear code, where N represents the number of users as well as messages and s represents the cardinality of side-information available at each user.
- One of the special cases of EICP is when it is specialized to cooperative data exchange problem. For such cases also, the normalized rate achieved in our case is $\frac{1}{\lceil \frac{s}{N-s} \rceil}$, if $s > \frac{N}{2}$ and $\frac{\lceil \frac{N-s}{s-1} \rceil}{1+\lceil \frac{N-s}{s-1} \rceil}$, if $s \leq \frac{N}{2}$. We prove that, when $(s-1)$ divides $(N-1)$ or $(N-s)$ divides $(N-1)$ or $s > \frac{2N+1-\sqrt{4N+1}}{2}$, this is less than the lower bound on the normalized rate, which is $\frac{M-s+1}{M}$, for scalar linear solutions to CDE problem [13].

The rest of the paper is organized as follows. The background and preliminaries are provided in Section II. In Section III, we define the specific class of EICP considered in this paper, namely, *Consecutive and Symmetric Embedded Index Coding Problem (CS-EICP)*. Our main result is summarized in the same section. Comparison of our results with the prior works is also done in the same section. The proof of this result is deferred to Section IV. Section V concludes this paper.

Notations: The finite field with q elements is denoted by $\mathbb{F}_q$. The set of all integers is denoted by $\mathbb{Z}$. $[n]$ represents the set $\{1, 2, \ldots, n\}$. $[a, b]$ represents the set $\{a, a+1, \ldots, b\}$, and $(a, b]$ represents the set $\{a+1, \ldots, b\}$. The bit wise exclusive OR (XOR) operation is denoted by $\oplus$. $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x. $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. All the message indices are taken modulo M while the user indices are taken modulo N. $a|b$ implies a divides b and $a \nmid b$ implies a does not divide b, for integers a and b.

II. Background and Preliminaries

Consider a system consisting of N users

$$S = \{S_0, S_1, \ldots, S_{N-1}\}$$

and M messages of d bits each,

$$X = \{x_0, x_1, ..., x_{M-1}\}, x_l \in \mathbb{F}_{2^d}, \forall l \in [0, M-1].$$

Let $K_j \subseteq X$ represent the subset of messages held by the user S_j and $W_j \subseteq X$ represent the subset of messages demanded by the user S_j, $j \in [0, N-1]$. We assume that $\cup_{j \in [0,N-1]} K_j = X$. Each user S_j broadcasts a set of y_j coded symbols each of size $d_1 = \frac{d}{z}$ bits, for some $z \in \mathbb{Z}$. Let $\mathcal{Y}_j, j \in [0, N-1]$, represent the set of all coded symbols transmitted by the user S_j,

$$\mathcal{Y}_j = \cup_{i=1}^{y_j} Y_j^i, \ : Y_j^i \in \mathbb{F}_{2^{d_1}},$$

where $Y_j^i, i \in [y_j]$, represents the i^{th} coded symbol of length d_1 bits, transmitted by the user S_j.

The *embedded index coding problem (EICP)* [6] is to minimize the number of bits broadcast by all users such that each user gets all the messages they have demanded, from the messages available with them and the coded symbols broadcast by the other users. That is, to minimize the *normalized rate*, which is defined as the total number of bits broadcast by all the users together normalized by the total bits of all the messages.

The decoding function, for embedded index coding problem, associated with some user S_j, is of the form

$$D_j : \{\cup_{i \in \{[0,N-1] \setminus j\}} \mathbb{F}_{2^{y_i d_1}}, \mathbb{F}_{2^{|K_j| d}}\} \rightarrow \mathbb{F}_{2^{|W_j| d}}.$$

III. Consecutive and Symmetric Embedded Index Coding Problem

In this section, we define the specific class of EICP considered in this paper, in Definition 1. We summarize our key result subsequently in Theorem 1. The proof of Theorem 1 is provided in Section IV. We compare our results with that in [6], [12] and [13]. We also illustrate our results using some examples.

Definition 1. ***Consecutive and Symmetric Embedded Index Coding Problem (CS-EICP)****: An EICP is said to be Consecutive and Symmetric Embedded Index Coding Problem if*

the side-information of each user $S_j, j \in [0, N-1]$, can be expressed as
$K_j = \{x_{(j+a) \ mod \ M}, x_{(j+a+1) \ mod \ M}, ..., x_{(j+a+s-1) \ mod \ M}\}$, *for some $a \in [0, M-1], s \in [1, M]$.*

A. Main Result

Without loss of generality, let the side-information set of each user $S_j, j \in [0, N-1]$, for CS-EICP, be $K_j = \{x_j, x_{(j+1) \text{ mod } M}, \dots, x_{(j+s-1) \text{ mod } M}\}$, for some $s \in [1, M]$.

Theorem 1. *For any CS-EICP, with $M = N$, $s \in [2, N-1]$, and demand set of each user $S_j, j \in [0, N-1]$, expressed as $W_j \subseteq X \backslash K_j$, the following normalized rate is achievable by using sub-packetization:*

$$\mathcal{C}(s) = \begin{cases} \frac{1}{\lceil \frac{s}{N-s} \rceil}, & \text{if } s > \frac{N}{2}. \\ \frac{\lceil \frac{N-s}{s-1} \rceil}{1 + \lceil \frac{N-s}{s-1} \rceil}, & \text{otherwise.} \end{cases} \quad (1)$$

B. Comparison with the results in [6] and [12]

In [6], a heuristic algorithm, which provides a scalar linear solution for the EICP, had been provided which involves calculating computationally complex min-rank of a graph. In [12], a scalar linear code achieving the length $N - s + 1$ was constructed explicitly in contrast to the computationally complex algorithm presented in [6] to find a scalar linear solution. We prove in Theorem 2 that for some range of values of s, the normalized rate achieved in our scheme, as in Theorem 1, using sub-packetization is lower than the normalized rate achieved in [12] .

Theorem 2. *For any CS-EICP, with $M = N$, and demand set of each user $S_j, j \in [0, N-1]$, expressed as $W_j \subseteq X \backslash K_j$, when $(s-1)|(N-1)$ or $(N-s)|(N-1)$ or $\frac{2N+1-\sqrt{4N+1}}{2} < s < N$, the normalized rate achieved in our scheme, as in Theorem 1, using sub-packetization is lower than the normalized rate $\frac{N-s+1}{N}$ achieved in [12] using scalar linear index code.*

The proof of Theorem 2 is provided in Section III in the arxiv version [14].

Remark 1. *For those ranges of values of s which are not discussed in Theorem 2, i.e., when $(s-1) \not| (N-1)$, $(N-s) \not| (N-1)$ and $\frac{N}{2} < s \leq \frac{2N+1-\sqrt{4N+1}}{2}$, we conjecture that the normalized rate achieved in our scheme, as in Theorem 1, using the idea of sub-packetization is lower than the normalized rate $\frac{N-s+1}{N}$ achieved in [12] using scalar linear index code.*

Remark 2. *One of the special cases of EICP, when the users demand all the messages which are not available as side-information, was studied as Cooperative Data Exchange (CDE) problem in [13]. A lower bound on the minimum number of transmissions, provided in [13] for the case when all the users have the same number of messages, i.e. s, as side-information, is $M - s + 1$, i.e., the normalized rate is lower bounded by $\frac{M-s+1}{M}$. If our scheme is specialized to CDE problem, when $(s-1)|(N-1)$ or $(N-s)|(N-1)$ or $\frac{2N+1-\sqrt{4N+1}}{2} < s < N$, the normalized rate achieved in our scheme, as in Theorem 1, using sub-packetization is lower than the lower bound on the normalized rate provided in [13] (as proved in Theorem 2).*

The following examples illustrate Theorem 1 and also the idea of sub-packetization that is invoked in the proof.

Example 1. *Let $N = 5, M = 5, s = 3$. Thus we have five messages $\{x_0, x_1, x_2, x_3, x_4\}$, each of size d bits, and five users $\{S_0, S_1, S_2, S_3, S_4\}$. Let the side-information set and the demand set corresponding to each user $S_j, j \in [0, 4]$, be $K_j = \{x_j, x_{(j+1) \ mod \ 5}, x_{(j+2) \ mod \ 5}\}$ and $W_j = \{x_{(j+3) \ mod \ 5}\}$ respectively.*

$$K_0 = \{x_0, x_1, x_2\} \quad K_1 = \{x_1, x_2, x_3\} \quad K_2 = \{x_2, x_3, x_4\}$$
$$K_3 = \{x_3, x_4, x_0\} \quad K_4 = \{x_4, x_0, x_1\}$$

$$W_0 = \{x_3\} \qquad W_1 = \{x_4\} \qquad W_2 = \{x_0\}$$
$$W_3 = \{x_1\} \qquad W_4 = \{x_2\}$$

We split each message into two disjoint blocks each of size $\frac{d}{2}$ bits, i.e.,

$$x_0 = \{x_0^0, x_0^1\} \qquad x_1 = \{x_1^0, x_1^1\} \qquad x_2 = \{x_2^0, x_2^1\}$$
$$x_3 = \{x_3^0, x_3^1\} \qquad x_4 = \{x_4^0, x_4^1\}$$

The coded symbols transmitted are linear combinations of these blocks. Each user $S_h, h \in [0, 4]$, transmits one coded symbol Y_h which includes 2 messages taken at an interval of 2. The 0^{th} block of the first message is taken while the 1^{st} block of the second message is taken. That is, for each $h \in [0, 4]$, the user S_h transmits $Y_h = x_h^0 \oplus x_{(h+2) \ mod \ 5}^1$. The transmitted coded symbols are

$$Y_0 = x_0^0 \oplus x_2^1 \qquad Y_1 = x_1^0 \oplus x_3^1 \qquad Y_2 = x_2^0 \oplus x_4^1$$
$$Y_3 = x_3^0 \oplus x_0^1 \qquad Y_4 = x_4^0 \oplus x_1^1.$$

Now, each user S_j needs to retrieve the demanded message $x_{(j+3) \ mod \ 5}$. Let us first consider the user S_0. The user S_0 retrieves x_3^0 from Y_3 since x_0 is available as side-information while it retrieves x_3^1 from Y_1. The user S_0 has decoded the message x_3 since it has retrieved all the blocks corresponding to the message x_3. Similarly all other users can decode their demanded message. Table I illustrates the coded symbols transmitted by each user and the coded symbols from which each user retrieves all the blocks corresponding to the demanded message. It can be noted from Table I that each user transmits $\frac{d}{2}$ bits owing to a normalized rate of $\frac{1}{2}$. The minimum number of bits required to transmit is $3d$ bits in [6], [12] while we were able to reduce it to $2.5d$ bits by utilizing the sub-packetization.

Example 2. *Let us take an example for the case when $s \leq \frac{N}{2}$ in Theorem 1. Let $N = M = 4, s = 2$ and the set of all messages and users be $\{x_0, x_1, x_2, x_3\}$ and $\{S_0, S_1, S_2, S_3\}$ respectively. Let the side-information set and the demand set corresponding to each user $S_j, j \in [0, 3]$ be $K_j =$*

Server S_i	Coded symbols transmitted by S_i	Message demanded by S_i: x_j	Message blocks-x_j^i of x_j	Coded symbols from which the x_j^i are decoded by S_i
S_0	$Y_0 = x_0^0 \oplus x_2^1$	x_3	x_3^0	Y_3
			x_3^1	Y_1
S_1	$Y_1 = x_1^0 \oplus x_3^1$	x_4	x_4^0	Y_4
			x_4^1	Y_2
S_2	$Y_2 = x_2^0 \oplus x_4^1$	x_0	x_0^0	Y_0
			x_0^1	Y_3
S_3	$Y_3 = x_3^0 \oplus x_0^1$	x_1	x_1^0	Y_1
			x_1^1	Y_4
S_4	$Y_4 = x_4^0 \oplus x_1^1$	x_2	x_2^0	Y_2
			x_2^1	Y_0

TABLE I
TABLE THAT ILLUSTRATES THE DECODING DONE BY EACH SERVER IN EXAMPLE 1

$\{x_j, x_{(j+1) \ mod \ 4}\}$ *and* $W_j = \{x_{(j+2) \ mod \ 4}\}$ *respectively.*

$$K_0 = \{x_0, x_1\} \qquad K_1 = \{x_1, x_2\}$$
$$K_2 = \{x_2, x_3\} \qquad K_3 = \{x_3, x_0\}$$

$$W_0 = \{x_2\} \quad W_1 = \{x_3\} \quad W_2 = \{x_0\} \quad W_3 = \{x_1\}$$

We split each message into three blocks of equal sizes, $x_j = \{x_j^0, x_j^1, x_j^2\}, j \in [0, 3]$.

$$x_0 = \{x_0^0, x_0^1, x_0^2\} \qquad x_1 = \{x_1^0, x_1^1, x_1^2\}$$
$$x_2 = \{x_2^0, x_2^1, x_2^2\} \qquad x_3 = \{x_3^0, x_3^1, x_3^2\}$$

The coded symbols transmitted are linear combinations of these blocks. For this case, the coded symbols are obtained in 4 *iterations. For each iteration* $h \in [0, 3]$*, the users* S_h *and* $S_{(h+1) \ mod \ 4}$ *are involved in the transmissions, where the coded symbols obtained by each user is by taking the first and the last messages available at each user. The user* S_h *transmits one coded symbol* Y_h^0 *where the* 0^{th} *block of the first message and the* 1^{st} *block of the last message available as side information are taken to be included in the coded symbol. The user* $S_{(h+1) \ mod \ 4}$ *transmits one coded symbol* Y_h^1*, where the* 1^{st} *block of the first message and the* 2^{nd} *block of the last message available as side information are taken to be included in the coded symbol. That is, for each* $h \in [0, 3], i \in [0, 1]$*, the user* $S_{(h+i) \ mod \ 4}$ *transmits* $Y_h^i = x^i_{(h+i) \ mod \ 4} \oplus x^{i+1}_{(h+i+1) \ mod \ 4}$*. The coded symbols transmitted are given below.*

$$Y_0^0 = x_0^0 \oplus x_1^1 \qquad Y_0^1 = x_1^1 \oplus x_2^2 \qquad Y_1^0 = x_1^0 \oplus x_2^1$$
$$Y_1^1 = x_2^1 \oplus x_3^2 \qquad Y_2^0 = x_2^0 \oplus x_3^1 \qquad Y_2^1 = x_3^1 \oplus x_0^2$$
$$Y_3^0 = x_3^0 \oplus x_0^1 \qquad Y_3^1 = x_0^1 \oplus x_1^2$$

Now, each user S_j *needs to retrieve the demanded message* $x_{(j+2) \ mod \ 4}$*. Let us first consider the user* S_0*. The user* S_0 *retrieves* x_2^0 *from* $Y_2^0 \oplus Y_2^1 = x_2^0 \oplus x_0^2$ *since* x_0 *is available as side-information while it retrieves* x_2^1 *and* x_2^2 *from* Y_1^0 *and* Y_0^1 *respectively. The user* S_0 *has decoded the message* x_2 *since it has retrieved all the blocks corresponding to the message* x_2*. Similarly all other users can decode their demanded message.*

Here the total number of bits transmitted by all the users together is $\frac{8d}{3}$ *bits which is less than* $3d$ *bits required to transmit in [6], [12].*

IV. PROOF OF THEOREM 1

In this section, we prove the achievability of Theorem 1 by providing a sub-packetization scheme. We split this problem into two disjoint cases depending on the value of s. We construct code for the two cases separately in the coming subsections. The proposed achievable schemes in both cases involve splitting the messages and transmitting their linear combination.

We split each message into z blocks, $x_l = \{x_l^0, x_l^1, \ldots, x_l^{z-1}\}, l \in [0, N-1]$. The value of z is given later in the coming subsections. We assume that d is sufficiently large such that this splitting of message into z blocks of equal sizes is possible. The size of each block is $d_1 = \frac{d}{z}$ bits. Each block is from a finite field $\mathbb{F}_{2^{d_1}}$. Each user transmits a linear combination of these blocks rather than the linear combination of the entire messages. All the users should be able to retrieve all the blocks corresponding to the demanded messages.

A. Case A: $s > \frac{N}{2}$.

In this subsection, we provide an achievable scheme for Case A.

Let $z = \left\lceil \frac{s}{N-s} \right\rceil$. We split each message into z blocks, $x_l = \{x_l^0, x_l^1, \ldots, x_l^{z-1}\}, l \in [0, N-1]$. The coded symbols transmitted are linear combinations of these blocks. Now, we provide the code construction.

Construction 1. *Each user* $S_j, \forall j \in [0, N-1]$*, transmits one coded symbol* Y_j*, where*

$$Y_j = \bigoplus_{k \in [0, z-1]} x^k_{(k(N-s)+j) \ mod \ N}$$

Each user S_j transmits one coded symbol Y_j which includes z messages taken at an interval of $(N-s)$. Also, z different blocks of these z messages are chosen, i.e., 0^{th} block of the first message is taken, 1^{st} block of the second message and so on. Since each of the messages in $\{\cup_{k \in [0, z-1]} x_{(k(N-s)+j) \text{ mod } N}\}$ is available with the user S_j, the coded symbol Y_j can be transmitted by S_j.

We need to establish that all the users are capable of retrieving all the demanded messages from the coded symbols obtained by Construction 1 and the side-information.

Proof of Decoding: Now, we prove that each user $S_j, j \in [0, N-1]$, can retrieve each of its demanded message $x_l \in W_j, l \in [0, N-1] \backslash [j, (j+s-1) \text{ mod } N]$.

It can be noted from Construction 1 that in any coded symbol $Y_{l'}$, for some $l' \in [0, N-1]$, if we take any block of a message present in $Y_{l'}$ which is needed by some user $S_h, h \in [0, N-1]$, then it can safely retrieve that block from $Y_{l'}$ since all other blocks in $Y_{l'}$ are available as side-information for the user S_h. This is since all the z

messages included in $Y_{l'}$ are taken at an interval of $N-s$ and $(z-1)(N-s) < s$(since $z = \left\lceil \frac{s}{N-s} \right\rceil$). Therefore, for each $i \in [0, z-1]$, the user S_j can retrieve x_l^i from $Y_{l'}$, where $l' = ((l-(N-s)i) \bmod N)$ as $(i+1)^{th}$ message chosen to be included in $Y_{l'}$ is x_l and i^{th} block of x_l is chosen.

$$\begin{aligned} Y_{l'} &= \bigoplus_{k\in[0,z-1]} x^k_{(k(N-s)+l') \bmod N} \\ &= x_{(i(N-s)+l') \bmod N} \bigoplus_{k\in[0,z-1]\setminus i} x^k_{(k(N-s)+l') \bmod N} \\ &= x_l^i \underbrace{\bigoplus_{k\in[0,z-1]\setminus i} x^k_{(k(N-s)+l') \bmod N}}_{\text{available as side-information}} \end{aligned}$$

B. Case B: $s \leq \frac{N}{2}$.

In this subsection, we provide an achievable scheme for Case B. Let $z = 1 + \left\lceil \frac{N-s}{s-1} \right\rceil$. We split each message into z blocks, $x_l = \{x_l^0, x_l^1, \ldots, x_l^{z-1}\}, l \in [0, N-1]$. The coded symbols transmitted are linear combinations of these blocks. The code construction for this case is given below.

Construction 2. *For each iteration $i \in [0, N-1]$,*

- *each user $S_{(k(s-1)+i) \bmod N}, k \in [0, z-2]$, transmits one coded symbol Y_k^i, where*

$$Y_k^i = x^k_{(k(s-1)+i) \bmod N} \oplus x^{k+1}_{((k+1)(s-1)+i) \bmod N}.$$

The coded symbols are obtained in N iterations. During each iteration $i \in [0, N-1]$, z messages at an interval of $s-1$ are chosen, $\left(\bigcup_{k\in[0,z-1]} x_{(k(s-1)+i) \bmod N}\right)$, and we make sure that z different blocks of these z messages are taken, i.e., 0^{th} block of the first message is taken, 1^{st} block of the second message and so on. And also, we choose $z-1$ users at an interval of $(s-1)$, i.e., $S_{(k(s-1)+i) \bmod N}, k \in [0, z-2]$, which are involved in the transmissions during iteration i, where the coded symbols obtained by each user is by taking the first and the last messages available at each user. Each user $S_{(k(s-1)+i) \bmod N}$ transmits one coded symbol Y_k^i where the k^{th} block of the first message $x_{(k(s-1)+i) \bmod N}$ and the $(k+1)^{th}$ block of the last message $x_{((k+1)(s-1)+i) \bmod N}$ available as side information are taken to be included in the coded symbol.

The proof that each user can decode their demanded messages using Construction 2 is provided in Section IV in the arxiv version [14].

Proof of Theorem 1: The total number of bits transmitted is $\frac{Nd}{\left\lceil \frac{s}{N-s} \right\rceil}$ bits for Case A. Hence, the normalized rate is $\frac{1}{\left\lceil \frac{s}{N-s} \right\rceil}$. The total number of bits transmitted is $\frac{\left\lceil \frac{N-s}{s-1} \right\rceil Nd}{1+\left\lceil \frac{N-s}{s-1} \right\rceil}$ bits for Case B. Hence, the normalized rate is $\frac{\left\lceil \frac{N-s}{s-1} \right\rceil}{1+\left\lceil \frac{N-s}{s-1} \right\rceil}$. This completes the proof.

V. CONCLUSION

In this paper, we have explored a specific classes of EICP, namely, consecutive and symmetric EICP. We have provided code construction for this case. By efficiently utilizing the subpacketization scheme, we were able to achieve a normalized rate lower than that of the state of the art [6], [12] for some cases. For other cases, we conjecture that the normalized rate achieved using our scheme is lower than that of the state of the art [6], [12]. In this paper, we had only explored a specific class of EICP. Explicit code construction for general EICP is still open. Exploring techniques to find a general solution is an interesting thing to work on.

ACKNOWLEDGEMENT

This work was supported by the Science and Engineering Research Board (SERB) of Department of Science and Technology (DST), Government of India, through J. C. Bose National Fellowship to B. Sundar Rajan. This work was done when Shanuja Sasi was at Purdue University as a visiting scholar under Science and Engineering Research Board (SERB) Overseas Visiting Doctoral Fellowship (OVDF).

REFERENCES

[1] Y. Birk and T. Kol, “Coding on demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients," in *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2825-2830, June 2006.

[2] M. Ji, A. M. Tulino, J. Llorca and G. Caire, “Caching and coded multi-casting: Multiple groupcast index coding," *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Atlanta, GA, 2014, pp. 881-885.

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

[4] A. Mazumdar, “On a duality between recoverable distributed storage and index coding," *2014 IEEE International Symposium on Information Theory*, Honolulu, HI, 2014, pp. 1977-1981.

[5] T. Luo, V. Aggarwal and B. Peleato, “Coded Caching With Distributed Storage," in *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 7742-7755, Dec. 2019.

[6] A. Porter and M. Wootters, “Embedded Index Coding," *2019 IEEE Information Theory Workshop (ITW),* Visby, Sweden, 2019, pp. 1-5.

[7] M. Li, L. Ong and S. J. Johnson, “Cooperative Multi-Sender Index Coding," in *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1725-1739, March 2019.

[8] C. Arunachala, V. Aggarwal and B. S. Rajan, “On the Optimal Broadcast Rate of the Two-Sender Unicast Index Coding Problem with Fully-Participated Interactions," in *IEEE Transactions on Communications*, vol. 67, no. 12, pp. 8612-8623, Dec. 2019.

[9] M. Li, L. Ong and S. J. Johnson, “Multi-Sender Index Coding for Collaborative Broadcasting: A Rank-Minimization Approach," in *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1452-1466, Feb. 2019.

[10] Z. MacHardy, A. Khan, K. Obana and S. Iwashina, “V2X Access Technologies: Regulation, Research, and Remaining Challenges," in *IEEE Communications Surveys & Tutorials,* vol. 20, no. 3, pp. 1858-1877, thirdquarter 2018.

[11] J. Pachat, N. S. Karat, Deepthi P P and B. S. Rajan, “Index Coding in Vehicle to Vehicle Communication," in *IEEE Transactions on Vehicular Technology*, 2020.

[12] A. A. Mahesh, N. S. Karat and B. S. Rajan, “Min-rank of Embedded Index Coding Problems," *2020 IEEE International Symposium on Information Theory (ISIT)*, Los Angeles, CA, USA, 2020, pp. 1723-1728.

[13] S. E. Rouayheb, A. Sprintson and P. Sadeghi, “On Coding for Cooperative Data Exchange," *2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo)*, Cairo, 2010, pp. 1-5.

[14] S. Sasi, V. Aggarwal and B. S. Rajan, “An Embedded Index Code Construction Using Sub-packetization," arXiv:2009.11329 [cs.IT], Oct 2020.