A Randomized Algorithm and Performance Bounds for Coded Cooperative Data Exchange

Alex Sprintson, Parastoo Sadeghi, Graham Booker, and Salim El Rouayheb

***Abstract*—We consider scenarios where wireless clients are missing some packets, but they collectively know every packet. The clients collaborate to exchange missing packets over an error-free broadcast channel with capacity of one packet per channel use. First, we present an algorithm that allows each client to obtain missing packets, with minimum number of transmissions. The algorithm employs random linear coding over a sufficiently large field. Next, we show that the field size can be reduced while maintaining the same number of transmissions. Finally, we establish lower and upper bounds on the minimum number of transmissions that are easily computable and often tight as demonstrated by numerical simulations.**

I. Introduction

The ever-growing demand of mobile wireless clients for large file downloads and video applications is straining cellular networks in terms of bandwidth and network cost. Inspired by the Internet paradigm where peer-to-peer (P2P) content delivery systems are more efficient than a server-client based model, one solution to address these issues is to allow the mobile clients to cooperate and exchange data directly among each other.

In this paper, we consider the problem of information exchange between a group of wireless clients. Each client initially holds a subset of packets and needs to obtain all the packets held by other clients. Each client can broadcast the packets in its possession (or a combination thereof) via a noiseless broadcast channel of capacity one packet per channel use. Assuming that clients can cooperate with each other and know which packets are available to other nodes, the aim is to minimize the total number of transmissions needed to satisfy the demands of all clients.

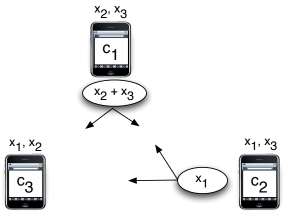

Fig. 1. Coded data exchange among three clients.

For example, Fig. 1 shows three wireless clients that are interested in obtaining three packets of m bits each, x_1, x_2 and $x_3 \in GF(2^m)$. The first, second and third clients have already obtained packets $\{x_2, x_3\}$, $\{x_1, x_3\}$ and $\{x_1, x_2\}$, respectively, i.e., each of these clients misses one packet. A simple cooperation scheme would consist of three uncoded transmissions. However, this is not an optimal solution since the clients can send coded packets and help multiple clients with a single transmission. The number of transmissions for this example can be decreased to two as shown in the figure.

The problem we consider may arise in many practical settings. For example, consider a wireless network in which some clients are interested in the same data (such as a popular video clip or an urgent alert message). Initially, the entire data is available at a base station and is broadcast to the interested clients. The communication link between the base station and the mobile clients can be, not only expensive and slow, but also unreliable or sometimes even non-existent, which causes some clients to receive only a portion of the data. Partial reception can be caused by channel fading or shadowing, connection loss, network saturation, or asynchronous client behavior such as in P2P systems. Despite this, whenever the whole data is collectively known by the interested clients, they can help each other to acquire the whole data using short-range client-to-client communication links or cooperative relaying which can be more affordable or reliable.

In this paper we investigate theoretical aspects of such client cooperation and are interested in finding efficient data exchange strategies which require minimum total number of transmissions. This problem was introduced in our preliminary work [1] where lower and upper bounds on the minimum number of transmissions were presented, in addition to a data exchange algorithm. We establish in this work new and improved lower and upper bounds. Furthermore, we propose an optimal data exchange algorithm based on random linear coding over

The first and third authors are with Texas A&M University, College Station, Texas, USA, Email: {spalex,gbooker}@tamu.edu. The second author is with the Australian National University, Canberra, ACT, Australia, Email: parastoo.sadeghi@anu.edu.au. The fourth author is with the University of California at Berkeley, Berkeley, California, USA, Email: salim@eecs.berkeley.edu. The work of Alex Sprintson was supported by NSP grant CNS-0954153 and by Qatar Telecom (Qtel), Doha, Qatar. The work of Parastoo Sadeghi was supported under Australian Research Council's Discovery Projects funding scheme (project no. DP0984950).

a large field and then show how coding can be performed over a smaller field, once the number of transmissions from each client is determined.

A closely-related problem is that of index coding [2]–[4] in which different clients cannot communicate with each other, but can receive transmissions from a server possessing all the data. Gossip algorithms [5] and physical layer cooperation [6] are also related concepts which are extensively studied in the literature.

II. System Model

Consider a set of n packets $X = \{x_1, \dots, x_n\}$ to be delivered to k clients belonging to the set $C = \{c_1, \dots, c_k\}$. The packets are elements of a finite alphabet which will be assumed to be a finite field $\mathbb{F}_q$ throughout this paper. At the beginning, each client knows a subset of packets denoted by $X_i \subseteq X$, while the clients collectively know all packets in X, i.e., $\cup_{c_i \in C} X_i = X$. We denote by $\overline{X}_i = X \setminus X_i$ the set of packets required by client c_i. We assume that each client knows the index of the packets that are available to other clients.

The clients exchange packets over a lossless broadcast channel with the purpose of making all packets in X available to all clients. The data is transferred in communication rounds, such that at round i one of the clients, say c_j, broadcasts a packet $p_i \in \mathbb{F}_q$ to the rest of the clients in C. Packet p_i may be one of the packets in X_j, or a combination of packets in X_j and the packets $\{p_1, \dots, p_{i-1}\}$ previously transmitted over the channel. Our goal is to devise a scheme that enables each client $c_i \in C$ to obtain all packets in $\overline{X}_i$ while minimizing the total number of transmissions. We focus on schemes that use linear coding over the field $\mathbb{F}_q$. As discussed in Section III below the restriction to linear coding operations does not result in loss of optimality.

With linear coding, any transmitted packet p_i is a linear combination of the original packets in X, i.e.,

$$p_i = \sum_{x_j \in X} \gamma_i^j x_j,$$

where $\gamma_i^j \in \mathbb{F}_q$ are the *encoding coefficients* of p_i. We refer to the vector $\gamma_i = [\gamma_i^1, \gamma_i^2, \dots, \gamma_i^n]$ as the *encoding vector* of p_i. The i-th *unit* encoding vector that corresponds to the original packet x_i is denoted by $u_i = [u_i^1, u_i^2, \dots, u_i^n]$, where $u_i^i = 1$ and $u_i^j = 0$ for $i \neq j$. We also denote by U_i the set of unit vectors that corresponds to the packets in X_i.

Let $n_i = |X_i|$ be the number of packets initially known to client c_i. The number of unknown packets to client c_i is therefore, $\bar{n}_i = |\overline{X_i}| = n - n_i$. We denote by $n_{\min} = \min_{1 \leq i \leq k} n_i$, the minimum number of packets known to a client. The corresponding client or clients form a subset $C_{\min}$ of C.

A client c_i is said to have a *unique* packet x_j if $x_j \in X_i$ and $x_j \notin X_\ell$ for all $\ell \neq i$. A unique packet can be broadcast by the client holding it in an uncoded fashion at any stage without any penalty in terms of optimality. Without loss of generality, we can assume that there are no unique packets in the system.

We note that the results of this paper can be applied, with minor modifications, to settings where the initial data available to clients include linear combinations of the packets in X. However, these settings are beyond the scope of this paper.

III. A Randomized Algorithm

In this section, we present a randomized algorithm for the data exchange problem. The algorithm operates over a finite field $\mathbb{F}_q$ of size $q > k \cdot n$ and identifies an optimal solution with probability at least $1 - \frac{nk}{q}$. The probability of success can be amplified by repeated application of the algorithm. We also show how to reduce the field size to $O(k)$ using bounds from the network coding literature.

A. Algorithm description and analysis

For clarity, we describe and analyze the algorithm in terms of encoding vectors, rather than original packets. That is, instead of saying that a packet $p_i = \sum_{x_j \in X} \gamma_i^j x_j$ has been transmitted, we say that we transmit the corresponding encoding vector $\gamma_i = [\gamma_i^1, \gamma_i^2, \dots, \gamma_i^n]$.

The algorithm operates in rounds. Assume that in round i, the encoding vector γ_i is transmitted by client $c_{t_i}, t_i \in \{1, \dots, k\}$. Then, the transmitted vector γ_i is a random linear combination of the unit vectors in U_{t_i}, i.e., $\gamma_i^j = 0$ for $x_j \notin X_{t_i}$; other elements of γ_i are selected at random from the field $\mathbb{F}_q$. The set $\Gamma_{i-1} = \{\gamma_1, \dots, \gamma_{i-1}\}$ contains the packets that have been transmitted during rounds $1, \dots, i-1$ of the algorithm. In general, the transmitted vector γ_i can be a linear function of the initial side information of c_{t_i} and the transmitted vectors in Γ_{i-1}. But since the vectors in Γ_{i-1} were received simultaneously by all the clients, there is no loss of generality in taking $\gamma_i \in \mathtt{span}(U_{t_i})$. The proofs of the correctness and the optimality of our algorithm, presented below, imply that this is optimal.

The formal description of the algorithm, referred to as *Random Data Exchange (RDE)*, appears on Fig. 2. The steps performed by the algorithm can be summarized as follows: At each iteration i, we select a client c_{t_i} with the highest rank of initial plus received encoding vectors up to the beginning of round i. That is,

$$t_i = \arg\max_{c_j \in C} \{\text{rank}(U_j \cup \Gamma_{i-1})\}; \qquad (1)$$

The chosen client c_{t_i} will then select a random linear combination of the packets in its *has* set which is then broadcast to all other clients. The process is

```
Algorithm RDE $(C, \{U_j, c_j \in C\}, \mathbb{F}_q)$:
  input:
    $C$ - set of clients
    $U_j$ - set of encoding coefficients available to client $c_j, j = 1, \ldots, k$
    $\mathbb{F}_q$ - the finite field

1  $i \leftarrow 1$
2  $\Gamma_0 \leftarrow \emptyset$
3  while there exists a client $c_j \in C$ for which it holds
   that $\mathrm{rank}(U_j \cup \Gamma_{i-1}) < n$ do
4      Select a client $c_{t_i}$ for which the set $U_{t_i} \cup \Gamma_{i-1}$ is
       of maximum rank, i.e.,
         $t_i = \arg\max_{c_j \in C} \{\mathrm{rank}(U_j \cup \Gamma_{i-1})\};$
5      Create a new encoding vector $\gamma_i$, such that $\gamma_i^j = 0$
       for $x_j \notin X_{t_i}$, otherwise $\gamma_i^j$ is a random element
       of field $\mathbb{F}_q$.
6      $\Gamma_i \leftarrow \Gamma_{i-1} \cup \{\gamma_i\}$
7      $i \leftarrow i + 1$
   endwhile
8  return $c_{t_1}, \ldots, c_{t_{i-1}}$ and $\gamma_1, \ldots, \gamma_{i-1}$
```

Fig. 2. Algorithm RDE

repeated until all clients possess n linearly independent combination of packets and hence, are able to obtain all the original packets in X.

We analyze the correctness and optimality of the algorithm. For each round i, we denote by OPT_i the minimal number of packets that still need to be transmitted after round i, i.e., in addition to the first i transmissions, in order to satisfy the demands of all the clients.

Consider iteration i of the algorithm. Let Q_{i-1} be an optimal set of encoding vectors required to complete the delivery of the packets to all clients after round $i-1$ has being completed. That is, Q_{i-1} includes OPT_{i-1} encoding vectors such that:

1) For each $\gamma \in Q_{i-1}$ it holds that $\gamma \in \mathtt{span}(U_j)$ for some $c_j \in C$;
2) For each client $c_j \in C$ it holds that the set $\Gamma_{i-1} \cup Q_{i-1} \cup U_j$ is of rank n.

Lemma 1: Let c_{t_i} be the client selected at Step 4 of the algorithm. Then, there exists at least one encoding vector, v, that can be removed from Q_{i-1} such that $\Gamma_{i-1} \cup (Q_{i-1} \setminus \{v\}) \cup U_{t_i}$ remains of rank n.

Proof: Let $\mu = \mathrm{rank}(U_{t_i} \cup \Gamma_{i-1})$ be the rank of the set of encoding vectors available to client c_{t_i}. Note that at the beginning of iteration i the rank of set $(U_{t_i} \cup \Gamma_{i-1})$ is at least as large as the rank of $(U_j \cup \Gamma_{i-1})$ of any other client $c_j \in C$. This implies that OPT_{i-1} is at least $n - \mu + 1$. Indeed, if there exists a client with strictly lower rank than μ, then this client would require at least $n - (\mu - 1)$ transmissions. Otherwise, if all clients have the same rank $\mu < n$, then the required number of transmissions is also at least $n - \mu + 1$ (Note that client c_{t_i} does not benefit from its own transmission at round i and hence, $OPT_{i-1} \geq 1 + (n - \mu)$. A similar argument is used in Lemma 8 in Section IV). Thus, there exists at least one encoding vector, v, that can be removed from Q_{i-1} such that $\Gamma_{i-1} \cup (Q_{i-1} \setminus \{v\}) \cup U_{t_i}$ remains of rank n. ■

Let v be a vector, whose existence is guaranteed by Lemma 1 at the end of round $i-1$. We denote by $\tilde{Q}_{i-1} = Q_{i-1} \setminus \{v\}$.

Note that for each client $c_j \in C \setminus \{c_{t_i}\}$ it holds that the rank of vector set $S_j = \Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_j$ is at least $n-1$. Let C' be as subset of $C \setminus \{c_{t_i}\}$ such that for each $c_j \in C'$ it holds that $\mathrm{rank}(S_j) = n-1$. Our goal is to show that vector γ_i, chosen randomly from the $\mathtt{span}(U_{t_i})$, increases the rank of each client $c_j \in C'$ with probability at least $1 - \frac{k}{q}$.

Let c_j be a client in C' and let ζ_j be the normal vector to the span of S_j, which is non-zero according to the definition of C'. Note that ζ_j can be written as

$$\zeta_j = \sum_{u_g \in U_{t_i}} \beta_g u_g + \sum_{u_g \in \overline{U}_{t_i}} \beta_g u_g,$$

where $\overline{U}_{t_i}$ is the set of unit encoding vectors that correspond to $\overline{X}_{t_i} = X \setminus X_{t_i}$.

Lemma 2: There exists $u_g \in U_{t_i}$ such that $\beta_g \neq 0$.

Proof: Suppose that it is not the case. Then, ζ_j can be expressed as $\zeta_j = \sum_{u_g \in \overline{U}_{t_i}} \beta_g u_g$. Then, ζ_j is a normal to $\mathtt{span}(U_{t_i})$. Since ζ_j is a normal to $\mathtt{span}(S_j)$ it is also normal to $\mathtt{span}(\Gamma_{i-1} \cup \tilde{Q}_{i-1})$. Thus, ζ_j is a normal to $\mathtt{span}(\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_{t_i})$ which contradicts the fact that $\mathrm{rank}\{\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_{t_i}\} = n$. ■

Let ζ'_j be a projection of ζ_j to $\mathtt{span}(U_{t_i})$, i.e., $\zeta'_j = \sum_{u_g \in U_{t_i}} \beta_g u_g$. Lemma 2 implies that ζ'_j is not zero.

Lemma 3: If for each client $c_j \in C'$ it holds that $\langle \zeta'_j, \gamma_i \rangle \neq 0$, then $OPT_i = OPT_{i-1} - 1$.

Proof: If the condition of the lemma is satisfied, then it holds that the rank of vector set $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_j \cup \{\gamma_i\}$ is n for all $c_j \in C$. This is because for clients $c_j \notin C'$, the rank of $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_j$ is n by definition, and for $c_j \in C'$, $\langle \zeta'_j, \gamma_i \rangle \neq 0$ implies that the rank of both $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_j \cup \{\gamma_i\}$ and $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_j \cup \zeta'_j$ is equal to n. Therefore, after iteration i of the algorithm, the data transfer can be completed within $OPT_{i-1} - 1$ rounds by transmitting the vectors in $\tilde{Q}_{i-1}$. ■

Recall that γ_i is a random linear combination of vectors in U_{t_i}, i.e., $\gamma_i = \sum_{u_g \in U_{t_i}} \gamma_i^g u_g$ where the γ_i^g's are i.i.d. random coefficients chosen from the field $\mathbb{F}_q$.

Lemma 4: For each client $c_j \in C'$, the probability that $\langle \zeta_j, \gamma_i \rangle$ is equal to zero is $\frac{1}{q}$.

Proof: The inner product $\langle \zeta_j, \gamma_i \rangle$ can be written as

$$\langle \zeta_j, \gamma_i \rangle = \sum_{u_g \in U_{t_i}} \beta_g \gamma_i^g. \qquad (2)$$

Let $\hat{U}$ be a subset of U_{t_i} such that for each $u_g \in \hat{U}$ it holds that $\beta_g \neq 0$. Lemma 2 implies that the set $\hat{U}$

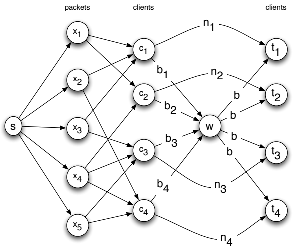

Fig. 3. Example multicast graph for 4 clients and 5 packets.

is not empty. Thus, $\langle \zeta_j, \gamma_i \rangle = \sum_{u_g \in \hat{U}} \beta_g \gamma_i^g$. Since the coefficients γ_i^g are i.i.d. uniformly distributed over $\mathbb{F}_q$, the probability that $\langle \zeta_j, \gamma_i \rangle$ is equal to zero is $\frac{1}{q}$. ■

Lemma 5: With probability at least $1-\frac{k}{q}$, it holds that $OPT_i = OPT_{i-1} - 1$.

Proof: Lemma 4 implies that for each client $c_j \in C'$, the probability that $\langle \zeta_j, \gamma_i \rangle$ is equal to zero is $\frac{1}{q}$. By using the union bound we can show that the probability that $\langle \zeta_j, \gamma_i \rangle = 0$ for some client $c_j \in C$ is bounded by $\frac{k}{q}$. Thus, with probability at least $1 - \frac{k}{q}$ it holds that $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_j \cup \{\gamma_i\}$ is of rank n for for every client $c_j \in C$. By Lemma 3, $OPT_i = OPT_{i-1} - 1$ with probability at least $1 - \frac{k}{q}$. ■

Theorem 6: The algorithm computes, with probability at least $1-\frac{k \cdot n}{q}$, an optimal solution for the data exchange problem, provided that the size q is larger than n.

Proof: Let OPT the be the optimum number of transmissions required to solve the data exchange problem. Note that $OPT_0 = OPT$. By Lemma 5, after each iteration, the number of required transmissions reduces by one with probability at least $(1 - \frac{k}{q})$. Thus, the data transfer will be completed after OPT iterations with probability at least

$$\left(1 - \frac{k}{q}\right)^{OPT} \geq \left(1 - \frac{k}{q}\right)^{n} \geq 1 - \frac{k \cdot n}{q},$$

where the last inequality holds for $q > n$. ■

By selecting a sufficiently large q (e.g., $q \geq 4k \cdot n$), we can guarantee a certain probability of success (e.g 3/4), which can then be amplified to be arbitrary close to 1 by performing multiple iterations and choosing the iteration that yields the minimum number of transmissions.

Corollary 7: For any $\varepsilon > 0$ the algorithm can find an optimal solution to the data exchange problem with probability at least $1 - \varepsilon$ in time polynomial in the size of the input and $\log(\varepsilon)$.

Reducing the field size

We can now construct a multicast problem as shown in Fig. 3 to reduce the required field size to $|\mathbb{F}_q| \geq k$. The multicast setting consists of a source node s and 4 layers. The first layer has n nodes corresponding to n source packets. The source node s is connected by a link to each node in layer 1. Layer 2 comprises k nodes corresponding to k clients. An existing edge $e_{\ell j}$ between node ℓ in the first layer and node j in the second layer means that client c_j knows packet x_ℓ. Client nodes in layer 2 are connected to a single node, w, in layer 3, where the edge capacity b_j represents the number of transmissions from c_j determined by the algorithm ($b_j = \sum_i 1[c_{t_i} = c_j]$, where $1[a]$ is the indicator function and becomes 1 only when condition a is true). And finally, w distributes coded packets to all k destination client nodes with edge capacities equal to $b = \sum_{j=1}^{k} b_j$. Obviously, client c_j is interested in all n source packets but also has side information X_j, which can also be represented by direct edges from the second to the last layer with capacities equal to n_j. This is a standard multicast problem of transmitting n packets from the source node s to k destinations. Using [7], we can find a network coding solution to the problem with $|\mathbb{F}_q| \geq k$.

We have thus shown that with linear coding we can achieve the optimal number of transmissions and achieve the capacity of the equivalent multicast problem. Hence, linear coding is sufficient for the data exchange problem.

IV. Lower and Upper Bounds

Before running the optimal randomized algorithm, the actual minimum number of transmissions OPT cannot be known a priori. It is therefore useful to be able to compute bounds on OPT. We first review one lower bound and one upper bound on OPT that were proved in [1]. We then establish some new bounds and comment on how they compare with previous bounds.

Lemma 8: [1] The minimum number of transmissions OPT satisfies $OPT \geq n - n_{\min}$. Moreover, if all clients initially have the same number of packets $n_{\min} < n$, then $OPT \geq n - n_{\min} + 1$.

Lemma 9: [1] For $|\mathbb{F}_q| \geq k$, the minimum number of transmissions OPT satisfies

$$OPT \leq \min_{1 \leq i \leq k} \{|\overline{X}_i| + \max_{1 \leq j \leq k} |\overline{X}_j \cap X_i|\}. \qquad (3)$$

The upper bound is obtained by making a client a *leader* with *uncoded* transmissions from other clients and then asking the leader to satisfy the demands of others.

Lemma 10: The minimum number of transmissions OPT satisfies

$$OPT \geq \left\lceil \frac{\sum_{i=1}^{k} \overline{n}_i}{k-1} \right\rceil = \left\lceil \frac{kn - \sum_{i=1}^{k} n_i}{k-1} \right\rceil, \qquad (4)$$

where $\lceil \cdot \rceil$ is the integer ceiling function.

Proof: The goal of transmissions is to reduce the total number of unknown packets from $\sum_{i=1}^{k} \overline{n}_i$ to zero. In each transmission round, the transmitting client cannot benefit from its own transmission. Therefore, at most $k-1$ clients will receive innovative information about their unknown packets. The lower bound follows by noting that the number of transmissions has to be an integer. ■

The next lower bound is a generalization of Lemma 8 and states that when there is at least one packet that is known only to clients with $n_{\min}$ known packets, the number of transmissions is at least $n - n_{\min} + 1$. Let $C_O = C \setminus C_{\min}$ be the set of clients such as c_i with $n_i > n_{\min}$. If C_O is non-empty, let $\overline{X}_O$ denote the set of common unknown packets for clients in C_O.

Lemma 11: Whenever $C_O = C \setminus C_{\min}$ is empty ($n_i = n_{\min} < n$ for all clients), then the minimum number of transmissions OPT satisfies $OPT \geq n - n_{\min} + 1$. When C_O is non-empty, we have $OPT \geq n - n_{\min} + \min(|\overline{X}_O|, 1)$, where $\overline{X}_O = \cap_{c_i \in C_O} \overline{X}_i$.
Compared with the Lemma 9 upper bound, here we wish to make the client c_i a leader (such that it acquires packets in $\overline{X}_i$) using *coded* transmissions from other clients called *helpers.*

Let vector $\mathbf{h}_j = \left(h_{j,1}, h_{j,2}, \cdots, h_{j,k_j}\right)$ denote the index of helper clients where k_j is the number of helpers and $h_{j,m}$ for $1 \leq m \leq k_j$ is the index of helpers. The transmission from helpers is done in the order of elements of $\mathbf{h}_j$. The overall index j in $\mathbf{h}_j$ refers to a particular choice of helpers. To make c_i a leader, we can at most have $(k-1)!$ distinct ordered helpers.

Helpers should collectively satisfy the demands of the leader: $\overline{X}_i \subseteq \cup_{m=1}^{k_j} X_{h_{j,m}}$. Each helper client $c_{h_{j,m}}$ is responsible for transmitting $A_{h_{j,m}} = |E_{h_{j,m}}|$ number of packets in its known set which are unknown to the leader and all previous helpers, where $E_{h_{j,m}}$ is defined as
$E_{h_{j,m}} = \overline{X}_i \cap \overline{X}_{h_{j,1}} \cap \overline{X}_{h_{j,2}} \cap \cdots \cap \overline{X}_{h_{j,m-1}} \cap X_{h_{j,m}}$.

Let $\Gamma_{j,m} = \{\gamma_{m,1}, \cdots, \gamma_{m,A_{h_{j,m}}}\}$ be the set of coding vectors transmitted by helper $c_{h_{j,m}}$ from its known set of unit vectors such that $\text{rank}(U_i \cup_{q=1}^{m} \Gamma_{j,q})$ increases by $A_{h_{j,m}}$ compared to $\text{rank}(U_i \cup_{q=1}^{m-1} \Gamma_{j,q})$. This is required to guarantee that c_i becomes the leader in the process and requires an appropriate choice of coding coefficients corresponding to unit vectors of $E_{h_{j,m}}$ for elements of $\Gamma_{j,m}$. Coefficients at other positions can be chosen at random from the chosen field $\mathbb{F}_q$ since they do not affect decoding at the leader.

After the helpers complete their transmissions, a client such as c_p will know $B_{j,p} = \text{rank}(U_p \cup_{m=1}^{k_j} \Gamma_{j,m})$ linearly independent equations and hence needs an extra $n - B_{j,p}$ transmissions from the leader. Summarizing the results, we arrive at the following upper bound:

Lemma 12: The minimum number of transmissions OPT satisfies

$$OPT \leq \min_{1 \leq i \leq k} \min_{j} \{|\overline{X}_i| + \max_{1 \leq p \leq k} \{n - B_{j,p}\}\}, \quad (5)$$

where $B_{j,p} = \text{rank}(U_p \cup_{m=1}^{k_j} \Gamma_{j,m})$, $\Gamma_{j,m} = \{\gamma_{m,1}, \cdots, \gamma_{m,A_{h_{j,m}}}\}$ is the set of coding vectors transmitted by helper $c_{h_{j,m}}$ from its known set of unit vectors $U_{h_{j,m}}$, and the second minimization is over *all or only a subset* of choices of helpers.

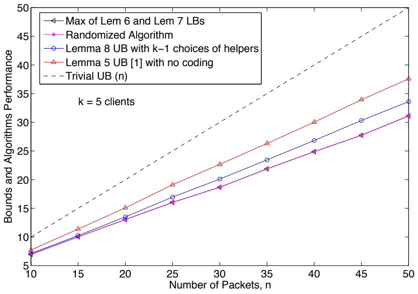

Fig. 4. Comparison of the derived lower and upper bounds with the optimal solution for $k = 5$ clients versus number of packets.

Finally, we present some numerical results. The bottom curve in Fig. 4 shows the maximum of the Lemma 10 and Lemma 11 lower bounds on OPT for $k = 5$ clients versus number of packets. The combined lower bounds provide very tight closed-form results on the optimal randomized solution which is also shown in the figure. The Lemma 12 upper bound is also shown using randomized coding with only $k-1$ choices of helpers for each leader. It significantly improves the Lemma 9 upper bound which used uncoded transmissions.

V. Conclusion

We presented a randomized algorithm for finding an optimal solution for the cooperative data exchange problem with high probability. While the algorithm gives a solution over a relatively large field, we showed that the field size can be reduced, through an efficient procedure, without any penalty in terms of the total number of transmissions. We also provided two tight lower bounds and one upper bound which can be easily computed and therefore, helpful in evaluating system performance.

In the future, we would like to explore two interrelated issues of (i) incentives and their overheads to guarantee continued cooperation from clients and (ii) fairness to clients (in terms of number of transmissions) during data exchange.

References

[1] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *Proc. Information Theory Workshop*, Cairo, Egypt, 2009, pp. 118–122.

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

[3] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Ko, "Index coding with side information," in *Proc. of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006, pp. 197–206.

[4] S. El Rouayheb, A. Sprintson, and C. N. Georghiades, "On the relation between the index coding and the network coding problems," *Proc. of IEEE International Symposium on Information Theory (ISIT)*, 2008.

[5] D. Shah, *Gossip Algorithms (Foundations and Trends in Networking)*. Now Publishers Inc, 2007, vol. 3, no. 1.

[6] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity–Part I: System description," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.

[7] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.