Pliable Index Coding

Siddhartha Brahma and Christina Fragouli

***Abstract*— We formulate a new variant of the index coding problem, where instead of demanding a specific message, clients are pliable, and are interested in receiving any t messages that they do not have. We term this problem pliable index coding or PICOD(t). We prove that, with this formulation, although some instances of the problem become simple, in general, the problem of finding the optimal linear code remains NP-hard. However, we show that it is possible to construct pliable index codes that are substantially smaller than index codes in many cases. If there are n clients, the server has m messages, and each client has a side information set of cardinality $s \leq m - t$; we show that $O(\min\{t \log n, t + \log^2 n\})$ broadcast transmissions are sufficient to satisfy all the clients. For $t = 1$, this is an exponential improvement over the n messages required in index coding in the worst case (for $m = n$). In addition, for $t \gg \log^2 n$, the number of broadcast transmissions required is only linearly dependent on t. We generalize the results to instances where the side information sets are not necessarily of equal cardinality. When $m = O(n^\delta)$, for some constant $\delta > 0$, we show that the codes of size $O(\min\{t \log^2 n, t \log n + \log^3 n\})$ are sufficient in general. We also consider the scenario when the server only knows the cardinality of the side information sets of the clients and each client is interested in receiving any t messages that it does not have. We term this formulation oblivious pliable index coding or OB-PICOD(t). If the cardinalities of side information sets of all the clients is s (with $s \leq m - t$), then we show that $\min\{s + t, m - s\}$ messages are both sufficient and necessary for linear codes. Finally, we develop efficient heuristic approximation algorithms for PICOD(t) and show through simulations on the random instances of PICOD(t) that they perform well in practice.**

***Index Terms*— Index coding, broadcast channel, greedy algorithm.**

I. Introduction

In THE well-known *Index Coding* with side-information problem, a server holds m messages, and can broadcast over a noiseless channel to a set of n receivers or clients. Each client has as side information some subset of the m messages, and requests from the server a *specific* message it does not have. The objective is to devise a coding strategy that minimizes the number of broadcast transmissions the server makes to satisfy the demands of all the clients [3]. Finding the smallest index code has been proven to be NP-Hard; in fact, it is even hard to find constant (multiplicative) factor approximations [4]–[6].

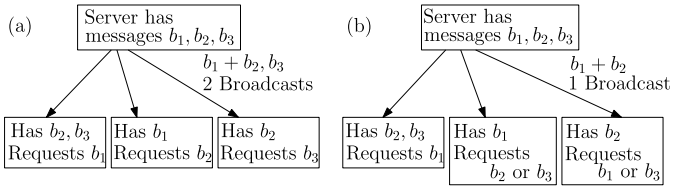

Fig. 1. (a) Index coding instance needs 2 broadcast transmissions. (b) PICOD(1) instance needs only one broadcast transmission.

But what if the clients are "pliable", and are interested in receiving *any single* message they do not have? There are several applications that motivate this relaxation. For example, several clients may be doing an internet search on the same topic (e.g. latest news), have collected some information and are interested in receiving with low delay, additional information on their topic of interest. As long as it is something they do not already have, they do not care which specific piece of information they receive. We term this formulation *Pliable Index Coding* (PICOD).

To illustrate the difference between index coding and pliable index coding, consider the scenario shown in Fig. 1. The server has three messages (in this case bits in the field $GF(2)$) b_1, b_2, b_3 and there are three clients with the side information sets $\{b_2, b_3\}$, $\{b_1\}$ and $\{b_2\}$ respectively. In a specific index coding instance, client 1 requests b_1, client 2 requests b_2 and client 3 requests b_3. For solving this, at least 2 broadcast transmissions (e.g. $b_1 + b_2$, b_3) are needed. Client 1 can decode b_1 from $b_1 + b_2$ as it knows b_2, where $+$ here denotes addition in the binary field $GF(2)$. Client 2 can decode b_2 from $b_1 + b_2$ as well, and client 3 gets b_3 directly. It is easy to see that one transmission does not suffice for this instance. In the case of PICOD, client 2 is interested in receiving any one of $\{b_2, b_3\}$ and client 3 is interested in receiving any one of $\{b_1, b_3\}$. Therefore, it is sufficient to send just $b_1 + b_2$ to satisfy all the clients. Clients 1 and 3 can decode $b_1 = b_2 + (b_1 + b_2)$ while client 3 can decode $b_2 = b_1 + (b_1 + b_2)$.

In this paper we restrict our discussion to *linear* codes for the pliable index coding problem. The first question we consider is – what is the complexity of computing the optimal (i.e. smallest) linear code for an instance of PICOD? Can it be done in polynomial time? Unfortunately, it turns out not to be the case; we prove that the problem of computing the optimal linear code for PICOD is NP-Hard.

Next, we quantify the number of transmissions required to solve an instance of PICOD. Consider the case where the side information sets of all the n clients are of the same cardinality s, with $s < m$. In index coding, when $m = n$ and $s = 0$ and each client requests a distinct message, n broadcast transmissions are required. Even for instances

Manuscript received September 12, 2014; revised May 30, 2015; accepted August 18, 2015. Date of publication September 10, 2015; date of current version October 16, 2015. This work was supported in part by the NOWIRE Project under Grant ERC-2009-StG-240317 and in part by the Research Project under Grant NSF-1527550. This paper, in parts, was presented at the IEEE International Symposium on Information Theory in 2012 [1] and the IEEE International Symposium on Information Theory in 2013 [2].

S. Brahma was with the Department of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland. He is now with the Department of Computer Science, University of Neuchâtel, Neuchâtel 2000, Switzerland (e-mail: sidbrahma@gmail.com).

C. Fragouli was with the Department of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland. She is now with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA 90095, USA (e-mail: christina.fragouli@ucla.edu).

Communicated by M. Langberg, Associate Editor for Coding Theory.
Digital Object Identifier 10.1109/TIT.2015.2477821

of the problem where the side information sets are random, Haviv and Langberg show that the minimum size of linear codes is $\Omega(\sqrt{n})$ with high probability. Clearly, pliable index coding can only do better, and much better in some cases. For example, when $s = 0$, exactly one broadcast transmission is sufficient to satisfy all the clients in PICOD. In fact, using a probabilistic argument we show that only $O(\log n)$ coded messages are sufficient in the worst case for *any* instance of PICOD where the cardinalities of the side information sets are all equal to s. That is, we prove an exponential improvement over index coding. When the cardinalities of the side information sets are arbitrary, we prove that a code of size $O(\min\{\log m(1+\log^+(\frac{n}{\log m})), m, n\})$ is sufficient, where $\log^+(x)$ denotes $\max\{\log x, 0\}$. To prove these bounds, we represent an instance of PICOD as a bipartite graph with vertices corresponding to messages and clients. The codes we construct, as described in Section V, correspond naturally to coverings of the graph with subgraphs where the degree of client vertices is one.

Next, we generalize PICOD to the case where each client requests *any t messages* it does not have (in our internet search example, each client requests t new pages). We denote this problem by PICOD(t) and hence PICOD simply corresponds to PICOD(1). In the case where the side information sets of all the clients are of the same cardinality $s \leq m - t$, a simple strategy would be to break it up into t instances or rounds of PICOD(1), where in each round every client learns one new message. This would result in $O(t \log n)$ coded broadcast transmissions from the server. However, when $t = m - s$, i.e., the clients request all the messages they do not have, a trivial solution using a MDS erasure correcting code [8] requires only $t = m - s$ transmissions, much less than $O(t \log n)$. Thus the simple strategy is clearly not optimal. In this paper we show that at most $O(\min\{t \log n, t + \log^2 n\})$ transmissions are required to solve this instance of PICOD(t). That is, if t is asymptotically larger than $\log^2 n$ and the cardinalities of the side information sets are all equal to s, then the number of broadcast transmissions needed grows *linearly* with t. We generalize the above results for PICOD(t) to the case where the side information sets of the clients are not necessarily of equal cardinality and where they are random.

The requirement of the server having full knowledge of the side information sets of the clients may be infeasible in certain situations, especially when feedback from the clients to the server is expensive. This motivates us to look at a different regime of pliable index coding where the server has very limited knowledge about the side information sets of the clients. Specifically, it only knows the cardinality but not the content of the side information sets. This formulation is reminiscent of the erasure channel formulation, where the source knows that a receiver has received a certain number of messages, but does not know which messages were lost. We call this problem *Oblivious Pliable Index Coding* or OB-PICOD. More generally, each client would like to know any t new messages it does not have and we denote this problem by OB-PICOD(t). In the case the side information sets of all the clients are of equal cardinality s, we show that to ensure all the clients receive at least t new messages (with $t \leq m - s$), the server needs to make at most $\min\{s + t, m - s\}$ broadcast transmissions using linear codes. We provide very simple constructions that achieve this bound. We also show that this is the best we can hope for using linear codes, by proving a matching lower bound on the size of the code.

The final contribution of this paper is the heuristic approximation algorithms we develop for PICOD(t). These are based on the bipartite graph representation of the instances and use a greedy approach to construct efficient codes. We also present two other algorithms specifically for PICOD(1). We show through extensive simulations on random instances that the greedy algorithm performs well in practice and that the size of the codes produced by the algorithm closely follows the behavior predicted by the theoretical upper bounds.

The remainder of the paper is organized as follows. After a brief overview of work related to index coding and its variants in Section II, in Section III we formally define the PICOD(t) and OB-PICOD(t) problems. In Section IV, we prove that finding the optimal linear code for PICOD is NP-Hard. In Section V, we prove upper bounds on the size of the codes required to solve an instance of PICOD(t) and in Section VI we prove upper and lower bounds on the size of the codes required to solve an instance of OB-PICOD(t). In Section VII, we develop efficient polynomial time heuristic approximation algorithms for solving PICOD(1) and PICOD(t). Finally, in Section VIII we show through extensive simulations on random pliable index coding instances that the algorithms work well in practice. We conclude with a discussion on possible future work in Section IX.

II. Related Work

Over the past few years, there has been a significant amount of work on the theory of index coding, especially for linear codes. The problem was introduced by Birk and Kol [3] in the context of an application in satellite communication networks. Bar-Yossef *et al.* [4] presented the first theoretical analysis of the problem. They showed that the optimal size for a *scalar linear* index code is given by the so called graph functional $minrk_2$. They conjectured this to be true even for non-linear codes, which was later disproved by Lubetzky and Stav [9].

By building on the works [6] and [10], which investigate the connections between index coding and network coding and by using information-theoretic linear programs, Blasiak *et al.* [11] prove some of the best known bounds for the index coding problem. The work of Blasiak *et al.* [11], [12] also shows several separation results between optimal linear and non-linear index codes. Techniques from interference alignment have also been used to analyze index codes [13]. In addition, there has been extensive work dealing with several aspects of the index coding problem including the complementary index coding problem [14], index codes with near extreme rates [15], secure index coding [16], index codes in presence of error [17] and index coding with outerplanar side information [18].

To the best of our knowledge, the problem of pliable index coding has not been examined before our work. However, there has been some work on other variations that differ

from the core index coding problem. The following are some representative examples. The so called bipartite index coding problem was analyzed in [19] and [20], where multiple clients may request a specific message. A special case of this problem, where the side information sets are of size one, was completely characterized in [21]. Finally, instantly decodable network codes were investigated in [22], where the clients request all the messages that they do not have and want them to be instantly decodable. This paper consolidates the results in our conference papers [1] and [2], and generalizes them for the case where there are an arbitrary number of clients and messages.

III. Problem Formulation

Suppose that the server has m messages $b_1, \cdots, b_m$ and there are n clients $c_1, \cdots, c_n$. The messages are assumed to lie in a suitably large finite field $(\mathcal{F}, +, \cdot)$ (whose size will be clarified later) and all the encoding and decoding functions are linear. For any positive integer x, let $[x]$ denote the set $\{1, \cdots, x\}$. For each $i \in [n]$, client c_i has as side information a subset of messages $b_{\sigma[i]}$, where $\sigma[i] \subset [m]$. Here $b_{\sigma[i]}$ denotes the set $\{b_j, j \in \sigma[i]\}$. Thus $\sigma[i]$ represents the indices of the messages that client c_i has as side information. Let t be the number of new messages each client requests. We assume that $|\sigma[i]| \leq m - t$ for all $i \in [n]$. The (linear) *Pliable Index Coding Problem* PICOD(t) is to construct a linear code $\mathcal{C}$ which consists of

1) A linear encoding function $E : \mathcal{F}^m \to \mathcal{F}^l$ mapping the messages $\mathbf{x} \in \mathcal{F}^m$, where l is the size of the code.
2) Linear decoding functions $D_1, \cdots, D_n$ for the n clients such that $D_i(E(\mathbf{x}), b_{\sigma[i]}) = \{b_{k_{i,1}}, \ldots, b_{k_{i,t}}\}$ for *some* t *distinct* indices $k_{i,1}, \ldots, k_{i,t} \in \overline{\sigma[i]} = [m] \setminus \sigma[i]$.

The goal is to find a code with the minimum size which is also the number of broadcast transmissions.

The *Oblivious Pliable Index Coding Problem* OB-PICOD(t) models a situation where the server has limited knowledge about the side information sets $\sigma[i]$. More concretely, we assume that the server only knows the cardinalities of the side information sets $|\sigma[i]|$. The (linear) OB-PICOD(t) problem is to construct a linear code $\mathcal{C}$ which consists of

1) A linear encoding function $E : \mathcal{F}^m \to \mathcal{F}^l$ mapping the messages $\mathbf{x} \in \mathcal{F}^m$, where l is the size of the code.
2) Linear decoding functions $D_1, \cdots, D_n$ for the n clients such that $D_i(E(\mathbf{x}), b_{\sigma[i]}) = \{b_{k_{i,1}}, \ldots, b_{k_{i,t}}\}$ for *some* t *distinct* indices $k_{i,1}, \ldots, k_{i,t} \in \overline{\sigma[i]} = [m] \setminus \sigma[i]$.

Note that, since the server does not have the exact side information sets but only their cardinalities, an encoding scheme should be able to deal with all possible such sets.

In this paper, we use the following result extensively.

Lemma 1: For a large enough finite field $\mathcal{F}$, given m symbols in it, there exist $n_0 \leq m$ linear combinations of the symbols such that any subset of $m_0 \leq n_0$ symbols can be recovered from these linear combinations if the remaining $m - m_0$ symbols are known.

The result follows from using Cauchy matrices as the coefficient matrix of the linear combinations [23]. If $m - m_0$ symbols are known, the n_0 linear combinations reduces to a system of n_0 linear equations in $m_0 \leq n_0$ unknowns in $\mathcal{F}$. Since any square sub-matrix of a Cauchy matrix is non-singular, this system can be solved to recover the m_0 unknown symbols. It is sufficient to choose a finite field of size at least $2m$ to ensure the existence of Cauchy matrices [24]. Therefore, we will assume that the messages reside in a field $\mathcal{F}$ of size at least $2m$.

IV. PICOD Is NP-Hard

For given side information sets, the size of the optimal pliable index code cannot be worse than the size of the optimal index code. This is because the index code encodes for a specific set of requested messages, which is just one of the many configurations allowed in the pliable case. In this section, we show that finding the pliable index code of minimum size is an NP-Hard problem. This will be accomplished by reducing an instance of the MONOTONE-1in3-SAT problem to an instance of PICOD(1).

In the MONOTONE-1in3-SAT problem, given a 3-SAT instance ϕ with all variables in non-negated form, the problem is to decide whether there is a satisfying assignment such that exactly one variable is `True` in each clause of the formula. MONOTONE-1in3-SAT has been shown to be NP-Hard by Schaefer [25]. Suppose ϕ is made up of M variables $\alpha_1, \cdots, \alpha_M$ and N_0 clauses

$$\phi(\alpha_1, \cdots, \alpha_M) = \bigwedge_{i=1}^{N_0} (\alpha_{i,1} \vee \alpha_{i,2} \vee \alpha_{i,3}) \tag{1}$$

where clause i is a disjunction of the literals $\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}$. The precise reduction is shown in the following lemma.

Lemma 2: Given an instance ϕ of MONOTONE-1in3-SAT as defined above, an instance I_{ϕ,M,N_0} of PICOD(1) can be constructed in polynomial time, such that ϕ has a satisfying assignment if and only if there is a code of size 1 for I_{ϕ,M,N_0}.

Proof: Given the MONOTONE-1in3-SAT instance ϕ, consider an instance I_{ϕ,M,N_0} of PICOD(1) defined as follows:

1) There are N_0 clients c_i ($i \in [N_0]$) corresponding to the clauses, where c_i corresponds to the clause i.
2) There are M messages b_j ($j \in [M]$) corresponding to the literals, where message b_j corresponds to the literal α_j. We assume that the messages b_j lie in the field $GF(2)$.
3) The side information set for c_i consists of all the messages that *do not* correspond to the literals in clause i. That is

$$\sigma[i] = \{j : \text{literal } \alpha_j \text{ is not in clause } i\} \tag{2}$$

Therefore, $|\sigma[i]| = M - 3$ and $|\overline{\sigma[i]}| = 3$ for all $i \in [N_0]$.

Suppose there exists a linear code of size 1 that is a solution to I_{ϕ,M,N_0}. It is necessarily of the following form

$$C_0 = b_{j_1} + b_{j_2} \cdots + b_{j_s} \tag{3}$$

for some $j_1, \cdots, j_s \in [M]$. Let $J_s = \{j_1, \cdots, j_s\}$. Since every client c_i must be able to decode at least one message not in $b_{\sigma[i]}$ and there is only one coded message, $\forall i \in [N_0]$, there exists $j_{k_i} \in J_s$ such that $j_{k_i} \in \overline{\sigma[i]}$. Since the corresponding message has to be decodable by c_i, there can be at most one

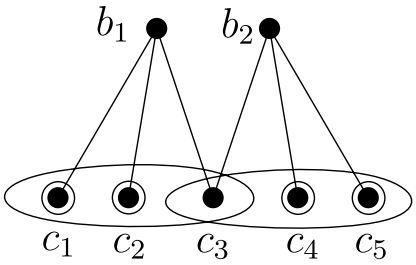

Fig. 2. Covering of the client vertices by neighboring message vertices. Here $B = \{b_1, b_2\}$ and $W_1(B) = \{c_1, c_2, c_4, c_5\}$.

such index. Thus, the set J_s has the property that exactly one of its members is present in each $\overline{\sigma[i]}$. Clearly, if we set the corresponding literals $\{\alpha_{j_k}, 1 \leq k \leq s\}$ to `True` and others to `False`, we make sure that all the clauses (which correspond to the clients) are satisfied and exactly one literal in each clause is `True`, which therefore satisfies ϕ. Thus, a code of size 1 for I_{ϕ,M,N_0} can be used to generate a satisfying assignment for ϕ that has exactly one `True` literal in each clause. Exactly the same argument can be reproduced backwards to prove the converse, which completes the reduction. Finally, it is easy to see that the reduction can be accomplished in polynomial time. ∎

Since MONOTONE-1in3-SAT is NP-Hard, Lemma 2 implies that computing the minimum size code in PICOD(1) is also NP-Hard. This also implies that computing the minimum size code for PICOD(t) in general is NP-Hard.

V. Upper Bounds for PICOD(t)

In this section, we prove upper bounds on the size of codes required for solving any instance of PICOD(t). We first consider the case of $t = 1$ and then generalize it.

We can visualize an instance of PICOD(1) using a bipartite graph G with m vertices on one side representing the messages (called "message vertices") and n vertices on the other side representing the clients (called "client vertices"). We identify the vertices by the messages or clients they represent. There is an edge from b_j to c_i if $j \in \overline{\sigma[i]}$, i.e., when message b_j is not in the side information set of client c_i. In Fig. 2 shown above, message b_1 is *not* in the side information sets of clients c_1, c_2, c_3 and hence is connected to them in G. In what follows, we denote the neighborhood of c_i in G by $N[c_i]$ and its degree by $d(c_i) = |N[c_i]|$. Similarly, $N[b_j]$ is the neighborhood of b_j in G and $d(b_j) = |N[b_j]|$ is its degree.

Consider two message vertices b_1 and b_2 and their neighborhoods $N[b_1]$ and $N[b_2]$ in G. We divide the client vertices in $N[b_1] \cup N[b_2]$ into two types, depending on the number of message vertices they are adjacent to. For a set of message vertices B, the set of client vertices that are adjacent to exactly one vertex in B is denoted by $W_1(B)$. In Fig. 2, for $B = \{b_1, b_2\}$, $W_1(B) = \{c_1, c_2, c_4, c_5\}$ and is depicted by the double circles. Note that if $b_1 + b_2$ is sent to these $|W_1(B)| = 4$ vertices, each of them can decode a message it does not have. Clients c_1 and c_2 can decode b_1 as they know b_2; similarly, c_4 and c_5 can decode b_2 as they know b_1. On the other hand, the set of clients which are adjacent to more than one message vertex, as is $\{c_3\}$, can decode neither b_1 nor b_2. The same logic can be extended if B contains more than two message vertices – if we transmit the sum of the messages in B, all the clients in $W_1(B)$ will be able to decode a message they do not have; in other words, it is sufficient to broadcast the sum message to satisfy all the $|W_1(B)|$ clients. We use this intuition to prove the following lemma.

Lemma 3: Without loss of generality, let $C = \{c_1, c_2, \ldots, c_k\}$ ($k \in [n]$) be any set of k client vertices and $d_{\max} = \max\{d(c_i) | i \in [k]\}$ and $d_{\min} = \min\{d(c_i) | i \in [k]\}$, where $d_{\min} \geq 1$. Assume that there is a fixed constant $r \geq 1$ such that $d_{\max} \leq r d_{\min}$. Then there is a code of size $O(\log(k))$ that satisfies all the clients in C.

Proof: We use a probabilistic argument to prove the lemma. Consider the neighborhood B_0 of C in G, i.e., $B_0 = \cup_{i=1}^{k} N[c_i]$. If $d_{\max} = 1$, then all the client vertices in C have degree 1, in which case the sum of all the messages in B_0 satisfies all the clients. Thus a code of size 1 is sufficient. Otherwise, if $d_{\max} \geq 2$, randomly select a subset B_1 of message vertices by selecting each vertex of B_0 with probability p (which will be determined later). Then the probability P_i of c_i being adjacent to exactly one vertex in B_1 is

$$P_i = d(c_i) p (1-p)^{d(c_i)-1} \tag{4}$$

The expected number of vertices in $W_1(B_1)$ is

$$E[|W_1(B_1)|] = \sum_{i=1}^{k} d(c_i) p (1-p)^{d(c_i)-1} \tag{5}$$

$$\geq k d_{\min} p (1-p)^{d_{\max}-1} \tag{6}$$

The expression $p(1-p)^{x-1}$ is maximized for $p = \frac{1}{x}$. Therefore by selecting $p = \frac{1}{d_{\max}}$ we get

$$E[|W_1(B_1)|] \geq k \frac{d_{\min}}{d_{\max}} \left(1 - \frac{1}{d_{\max}}\right)^{d_{\max}-1} \tag{7}$$

$$\geq k \frac{d_{\min}}{e d_{\max}} \geq \frac{k}{er} \tag{8}$$

Here e is Euler's constant. By the probabilistic method [26], there is at least one subset B_1 for which $|W_1(B_1)| \geq \frac{k}{er}$ which means the sum of the messages in B_1 can satisfy a constant fraction of the k client vertices. We are then left with at most $k(1-\frac{1}{er})$ client vertices. Since the set of message vertices does not change, the ratio of the maximum and minimum degrees in the remaining client vertices is also bounded by r and hence the above argument can be repeated until only a constant number of client vertices are left. Since the number of client vertices reduces by a constant fraction in each iteration, at most $O(\log(k))$ coded messages are required to satisfy all the k clients in C. ∎

In particular, if the cardinalities of the side information sets of all the clients are equal, a code of size $O(\log(n))$ is sufficient to satisfy all the clients. This is an exponential improvement over the $\Omega(n)$ messages required for index coding in the worst case for $m = n$. For a general instance of PICOD(1), where the cardinalities of the side information sets are arbitrary, we use a suitable partition of the client

vertices along with Lemma 3 to prove the following result. Here $\log^+(x)$ denotes $\max\{\log x, 0\}$.

Theorem 4: For any PICOD(1) instance with m message and n client vertices, all the client vertices can be satisfied with a code of size $O(\min\{\log m(1+\log^+(\frac{n}{\log m})), m, n\})$.

Proof: The degrees of the client vertices can range from 1 to m. Partition the vertices into sets S_i' such that $S_i' = \{c_l | 2^{i-1} \leq d(c_l) < 2^i\}$. For ease of notation, we only retain the non-empty ones and rename them to S_1 to S_g, where g is the number of non-empty S_i''s. Clearly the ratio of the maximum and minimum degrees in each of the sets S_i is at most $r = 2$ and $g \leq 1 + \lceil \log_2(m) \rceil$. We partition the sets further into two groups as follows. Let G_1 contain all the indices i such that $|S_i| < 3$ and let G_2 contain all indices such that $|S_i| \geq 3$. Since $\sum_{i \in G_2} |S_i| \leq \sum_{i \in [g]} |S_i| = n$, we have $|G_2| \leq n/3$. Also, $|G_1| + |G_2| = g$. By Lemma 3, we need at most $K_1(1 + \log(|S_i|))$ messages to satisfy the clients in S_i, for some absolute constant K_1. The additional "1" in the expression is to cover cases when $|S_i| = 1$. Therefore, the total number of coded messages required is at most

$$K_1 \sum_{i=1}^{g} (1 + \log(|S_i|)) \tag{9}$$

$$\leq K_1 \sum_{i \in G_1} (1 + \log(|S_i|)) + K_1 \sum_{i \in G_2} (1 + \log(|S_i|)) \tag{10}$$

$$\leq K_1 |G_1| (1 + \log(2)) + K_1 |G_2| + K_1 |G_2| \log\left(\frac{\sum_{i \in G_2} |S_i|}{|G_2|}\right) \tag{11}$$

$$\leq q K_1 |G_1| (1 + \log(2)) + K_1 |G_2| + K_1 |G_2| \log\left(\frac{n}{|G_2|}\right) \tag{12}$$

$$= O\left(\log m \left(1 + \log^+\left(\frac{n}{\log m}\right)\right)\right) \tag{13}$$

The second inequality follows from Jensen's inequality applied to the $\log(.)$ function and the fact that $|S_i| < 3$ for $i \in G_1$. The last can be argued as follows. Note that the function $x \log(n/x)$ is non-decreasing for $x \leq n/e$. Therefore, if $1 + \lceil \log m \rceil \leq n/3$, since $|G_2| \leq g$,

$$|G_2| \log(\frac{n}{|G_2|}) = O(\log m \log(n/\log m)) \tag{14}$$

Otherwise, if $n/3 < 1 + \lceil \log m \rceil$, since $|G_2| \leq n/3$,

$$|G_2| \log(\frac{n}{|G_2|}) \leq \frac{n}{3} \log(3) = O(\log m) \tag{15}$$

In addition, it is easy to see that all the clients can also be satisfied trivially by broadcasting all the messages (total of m) or broadcasting one message not in the side information set of each client (total of at most n). By combining all the three bounds, we obtain our claimed result. ∎

In particular, if the number of messages is polynomially related to the number of clients, i.e., $m = O(n^\delta)$ for some constant $\delta > 0$, then a code of size $O(\log^2 n)$ is sufficient for any instance of PICOD(1). Again, this is an almost exponential improvement over the worst case code size in index coding.

We now generalize the bounds for PICOD(1) to PICOD(t). Given a subset of message vertices B, we can categorize the client vertices in the neighborhood of B according to the number of message vertices they are adjacent to. The set of client vertices that are adjacent to exactly i message vertices in B, is denoted by $W_i(B)$. In Fig. 3a, for $B = \{b_1, b_2\}$, $W_1(B) = \{c_1, c_2, c_4, c_5\}$ and $W_2(B) = \{c_3\}$. As previously mentioned, a single linear combination is sufficient to satisfy all the client vertices in $W_1(B)$. Applying Lemma 1, two independent linear combinations of messages in B are sufficient to satisfy all client vertices in $W_2(B)$.

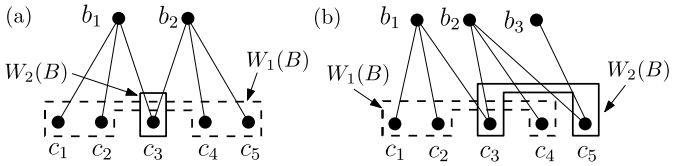

Fig. 3. Covering of the client vertices by neighboring message vertices. (a) Here $B = \{b_1, b_2\}$ and $W_1(B) = \{c_1, c_2, c_4, c_5\}$ and $W_2(B) = \{c_3\}$. (b) Here $B = \{b_1, b_2, b_3\}$ and $W_1(B) = \{c_1, c_2, c_4\}$ and $W_2(B) = \{c_3, c_5\}$.

In another example, consider the graph in Fig. 3b. In this case $B = \{b_1, b_2, b_3\}$, $W_1(B) = \{c_1, c_2, c_4\}$ and $W_2(B) = \{c_3, c_5\}$. If $b_1 + b_2 + b_3$ is sent to the 3 vertices in $W_1(B)$, each of them can decode a message it does not have. For the two vertices in $W_2(B)$, notice that they are adjacent to two different subsets of message vertices. In this case, we can apply Lemma 1 to obtain an additional linear combination of $\{b_1, b_2, b_3\}$ such that each of c_3, c_5 can decode two messages it does not have.

For any subset of message vertices B, define $t(B) = \max\{i | W_i(B) \neq \emptyset\}$. In general, Lemma 1 implies the following result.

Lemma 5: For any B there exists a set of $t(B)$ linear combinations of the messages in B such that each client vertex in $W_i(B)$ can decode i messages it does not have, for $i \in [t(B)]$.

To build smaller codes for PICOD(t), our approach is to select a suitable subset of message vertices such that a *large* number of client vertices have approximately t (or a constant fraction of t) message vertices adjacent to them. To this end, we use a probabilistic argument. Similar to Lemma 3, the following lemma proves an upper bound for the case when the side information sets have low variation in their cardinalities.

Lemma 6: Without loss of generality, let $C = \{c_1, c_2, \ldots, c_k\}$ ($k \in [n]$) be any set of k client vertices and $d_{\max} = \max\{d(c_i) | i \in [k]\}$ and $d_{\min} = \min\{d(c_i) | i \in [k]\}$, where $d_{\min} \geq t$. Assume that there is a fixed constant $r \geq 1$ such that $d_{\max} \leq r d_{\min}$. Then there is a code of size $O(\min\{t \log k, t + \log^2 k\})$ such that each client vertex can decode t messages it does not have.

Proof: If $k = 1$, then the only client in C can be satisfied by simply sending t distinct messages it does not have. Otherwise, if $k \geq 2$, randomly select a subset B_1 of message vertices by selecting each vertex with probability $p = \frac{t}{d_{\max}}$. Without loss of generality, consider a particular client vertex c_1 with degree d in the graph. Let X_i denote the indicator variable which is 1 if the i-th neighbor of c_1 is present in the sample B_1, for $i \in [d]$. Clearly, the X_i's are

i.i.d Bernoulli random variables with $P(X_i = 1) = p$. Let $X = X_1 + \ldots + X_d$. Then we have

$$E[X] = E[\sum_{i=1}^{d} X_i] = \sum_{i=1}^{d} E[X_i] = dp = \frac{dt}{d_{\max}} \quad (16)$$

Since $d \leq d_{\max} \leq rd$, $t \geq E[X] \geq \frac{t}{r}$. We also need concentration bounds for $E[X]$. Since X is a sum of i.i.d Bernoulli random variables, we can use the following Chernoff bounds which are valid for any $\epsilon \in (0, 1)$ [27].

$$P(X < (1-\epsilon)E[X]) \leq e^{-\frac{\epsilon^2}{2}E[X]} \quad (17)$$

and

$$P(X > (1+\epsilon)E[X]) \leq e^{-\frac{\epsilon^2}{3}E[X]} \quad (18)$$

Assume that $t \geq 24r \log k$. If we choose $\epsilon = \sqrt{\frac{6r \log k}{t}}$, then clearly $\epsilon \leq \frac{1}{2}$. Then, by using the fact that $E[X] \geq t/r$, we have

$$P(X > 3E[X]/2) \leq P(X > (1+\epsilon)E[X]) \leq e^{-\frac{\epsilon^2}{3}E[X]} \quad (19)$$

$$\leq e^{-\frac{6r \log k}{3t} \cdot \frac{t}{r}} = e^{-2 \log k} = k^{-2} \quad (20)$$

We can conclude that

$$P(X > 3E[X]/2) \leq k^{-2} \quad (21)$$

Similarly,

$$P(X < E[X]/2) \leq P(X < (1-\epsilon)E[X]) \leq e^{-\frac{\epsilon^2}{2}E[X]} \quad (22)$$

$$= e^{-\frac{6r \log k}{2t} \cdot \frac{t}{r}} = e^{-3 \log k} = k^{-3} \quad (23)$$

We can conclude that

$$P(X < E[X]/2) \leq k^{-3} \quad (24)$$

Combining the above two results, we have

$$P(\frac{E[X]}{2} \leq X \leq \frac{3E[X]}{2}) \geq 1 - \frac{1}{k^2} - \frac{1}{k^3} \quad (25)$$

which implies that with non-zero probability, a particular client vertex has between $E[X]/2$ and $3E[X]/2$ adjacent message vertices in B_1. Therefore, the expected number of client vertices having the same property is at least $k - 1/k - 1/k^2 \geq k - 3/4$, for $k \geq 2$. By the probabilistic method, there is at least one subset B_1 for which the expected value is reached or surpassed. This implies that there is a subset of message vertices B_1, such that all client vertices have between $E[X]/2$ and $3E[X]/2$ message vertices adjacent to them (here there is slight abuse of notation and $E[X]$ for each client may be different). By an application of Lemma 1, there is a set of at most $3E[X]/2 \leq 3t/2$ linear combinations of the corresponding messages such that each client vertex can decode at least $E[X]/2 \geq t/2r$ messages it does not have. Note that if $3t/2 > d_{\max}$, then the number of linear combinations can be capped to $d_{\max}$.

Therefore, if $t \geq 24r \log k$, by using $O(t)$ broadcast messages we can make sure all the client vertices learn at least $t/2r$ new messages. We can now recursively use the same argument for an instance where each client now needs (at most) $t' = t(1 - 1/2r)$ new messages. However, in the new instance, the ratio of the maximum and minimum degrees has changed and may become larger than a constant. Consider the case when $d_{\min} \geq 2t$. As client vertices get to know new messages, their degrees decrease. For $0 \leq t_1, t_2 \leq t$, the ratio of the maximum and minimum degree of client vertices in any instance can be upper bounded as follows.

$$\frac{d_{\max} - t_1}{d_{\min} - t_2} \leq r + \frac{rt_2 - t_1}{d_{\min} - t_2} \leq 2r \quad (26)$$

The last inequality follows from the fact that $t_2 \leq t$ and $d_{\min} \geq 2t$. In this case, the ratio does not exceed a constant $(2r)$ in the worst case. Therefore, it is possible to proceed with the recursion. If $d_{\min} < 2t$, since the client vertices have degree at most $d_{\max} \leq 2tr$, all of them can be satisfied by sending $O(t)$ linear combinations, without need for recursion.

In case the recursion continues, we stop it when t' becomes less than $24r \log k$. In this case, using Lemma 3 t' times, all the remaining clients can be satisfied by using $O(t' \log k)$ messages. Thus, if $f(k, t)$ is the number of messages required for sending t unknown messages to each of the k clients, we get the following recurrence

$$f(k,t) \leq \begin{cases} f(k, t(1 - \frac{1}{2r})) + O(t) & \text{if } t \geq 24r \log k \\ O(t \log k) & \text{otherwise} \end{cases} \quad (27)$$

This recurrence can be solved to obtain the claimed bound of $f(k, t) = O(\min\{t \log k, t + \log^2 k\})$. ∎

In particular, if the side information sets of all the clients are of equal cardinality, a code of size $O(\min\{t \log n, t + \log^2 n\})$ is sufficient to satisfy all the clients. In this case, when $t \gg \log^2 n$, the number of messages required grows linearly with t. Since at least t messages are required to solve any instance of PICOD(t) using linear codes, the bounds are tight within a constant multiplicative factor for $t \gg \log^2 n$. For the general case of client vertices having arbitrary degrees, we can use Lemma 6 to derive the following bound. As before, $\log^+(x)$ denotes $\max\{\log x, 0\}$.

Theorem 7: For any PICOD(t) instance with m message and n client vertices, all the client vertices can be satisfied with a code of size that is

$$O\left(\min\left\{\begin{array}{l} t \log m \left(1 + \log^+\left(\frac{n}{\log m}\right)\right), \\ t \log m + \log m \log^2 n, m, tn \end{array}\right\}\right) \quad (28)$$

Proof: The degrees of the client vertices can range from 1 to m. Partition the vertices into subsets S_i' such that $S_i' = \{c_l | 2^{i-1} \leq d(c_l) < 2^i\}$. For ease of notation, we only retain the non-empty ones and rename them from S_1 to S_g, where g is the number of non-empty S_i''s. Clearly the ratio of the maximum and minimum degrees in each of the sets S_i is at most $r = 2$ and $g \leq 1 + \lceil \log_2(m) \rceil$. By Lemma 6, we need at most the minimum of $K_1 t(1 + \log(|S_i|))$ and $K_2(t + \log^2(|S_i|))$ messages to satisfy the clients in S_i, for some absolute constants K_1, K_2. Using the first term and the proof of Thm. 4, the total number of coded messages required

can be upper bounded by

$$K_1 t \sum_{i=1}^{g}(1+\log(|S_i|)) = O\left(t \log m \left(1+\log^+\left(\frac{n}{\log m}\right)\right)\right) \tag{29}$$

Using the second term, the total number of coded messages required can be upper bounded by

$$K_2 \sum_{i=1}^{g}(t+\log^2(|S_i|)) \tag{30}$$

$$= K_2(tg + \sum_{i=1}^{g}\log^2(|S_i|)) \leq K_2(tg + g\log^2 n) \tag{31}$$

$$= O(t\log m + \log m \log^2 n) \tag{32}$$

Here we have used the fact that $|S_i| \leq n$. In addition, all the clients can be satisfied trivially by broadcasting all messages (m) or broadcasting t messages not in the side information set of each client (at most tn). Combining all the four bounds, we obtain our claimed result. ∎

In particular, if the number of messages is polynomially related to the number of clients, i.e., $m = O(n^\delta)$ for some constant $\delta > 0$, then a code of size $O(\min\{t\log^2 n, t\log n + \log^3 n\})$ is sufficient for any instance of PICOD(t). Note that the proof of Thm. 28 uses Lemma 1 as a key building block, so the choice of the field $\mathcal{F}$ of a suitable size is important. However, Thm. 4 is not dependent on the specific choice of $\mathcal{F}$.

For random instances of PICOD(t), we have the following result.

Theorem 8: For large enough m, if each message appears in the side information set of a particular client with a fixed and constant probability q, then any instance of PICOD(t) can be solved with a code of size $O(\min\{t\log n, t+\log^2 n\})$ with high probability.

Proof: By the law of large numbers, the degree of each client vertex in the bipartite graph representation of a random instance of PICOD(t) is concentrated near the mean $m(1-q)$. For any $\epsilon > 0$, for a large enough m, $d(c_i) \in [m(1-q-\epsilon), m(1-q+\epsilon)]$ with high probability. If we select an $\epsilon < q/3$, the ratio of the maximum and minimum degrees is ≤ 2. Then the claim follows from Lemma 6. ∎

VI. Tight Bounds for OB-PICOD(t)

In this section, we consider the OB-PICOD(t) problem. As formulated in Section III, in this problem the server only knows the cardinality of the side information sets of the clients and has to broadcast messages such that each client can decode t new messages it does not have. We denote the maximum size of the side information sets by $s_{\max}$ and the minimum size by $s_{\min}$. We assume $s_{\max} \leq m - t$. Also, let Σ denote the set of *distinct* cardinalities of the side information sets in an instance. Our main result is summarized in the following theorem.

Theorem 9: For any OB-PICOD(t) instance with m messages, if each client has as side information at least $s_{\min}$ and at most $s_{\max}$ messages, then the server needs to broadcast at most $\min\{s_{\max}+t, m-s_{\min}\}$ messages to satisfy all the clients using linear codes. When $s_{\max} = s_{\min} = s$, this bound is tight for linear codes.

We prove this result by combining two lemmas, Lemma 10 and Lemma 11, that we next state and prove. The first lemma, Lemma 10, gives a simple upper bound on the size of codes for OB-PICOD(t).

Lemma 10: To solve any instance of OB-PICOD(t), $\min\{s_{\max}+t, m-s_{\min}\}$ coded messages are sufficient using linear codes.

Proof: There are two ways to make sure that each client is able to decode at least one message that it does not have. We can select any $s_{\max}+t$ messages and send them uncoded, one at a time. Since the side information sets have size at most $s_{\max}$, there exists at least t messages that a client does not have. As another strategy, consider $m - s_{\min}$ linear combinations of all the messages obtained by the application of Lemma 1. Since each client knows at least $s_{\min}$ messages, it can recover at least t (in fact all) messages it does not know. Taking the minimum of the two strategies, we get an upper bound of $\min\{s_{\max}+t, m-s_{\min}\}$ messages. ∎

In particular when $s_{\min} = s_{\max} = s$, $\min\{s+t, m-s\}$ coded messages are sufficient to solve an instance of OB-PICOD(t). If all the clients have very large side information sets ($s_{\min}$ is close to m) or very small side information sets ($s_{\max}$ is small), the server can satisfy all the clients using only t plus a small number of broadcasts, without even knowing the exact side information sets.

We now prove lower bounds for OB-PICOD(t) for linear codes. Let $\mathbf{e}_1, \ldots, \mathbf{e}_m$ be the unit vectors in $\mathcal{F}^m$, i.e., $\mathbf{e}_i$ has a 1 (the identity element in $\mathcal{F}$) in the i-th position and 0 (the zero element in $\mathcal{F}$) elsewhere. Thus they form an orthogonal basis for $\mathcal{F}^m$. Since the server uses linear encodings, the j-th encoded message can be represented as a dot product of an encoding vector $\mathbf{a}_j$ and the message vector $\mathbf{b} = \{b_1, \ldots, b_m\}$, with operations done over the field $\mathcal{F}$. For l encoded messages, we have the corresponding l encoding vectors $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_l\}$. We denote the vector space spanned by the vectors in A by $Span(A)$. For a subset of indices $U \subseteq [m]$, e_U denotes the corresponding set of unit vectors $\mathbf{e}_i, i \in U$.

Lemma 11: For linear codes, any scheme for OB-PICOD(t) needs at least $\max_{s\in\Sigma} \min\{s+t, m-s\}$ messages.

Proof: Consider a particular client whose side information set is of size $s \in \Sigma$. The knowledge of the side information set can be expressed equivalently by the fact that the client can compute any vector in the span of $S = \{\mathbf{e}_{\alpha_1}, \ldots, \mathbf{e}_{\alpha_s}\}$ where $\alpha_1, \ldots, \alpha_s$ are the indices of the messages that is in its side information set. In the case of linear codes, for a client to be able to decode the i-th message using its side information sets and the encoded messages it is easy to see that $\mathbf{e}_i$ should belong to the span of $A \cup S$ (for a proof see [4]).

$$\mathbf{e}_i \in Span(A \cup S) \tag{33}$$

Let the union of all unique message indices that are decoded by all the clients be U. We claim that $|U| \geq s+t$. Indeed, if $|U| < s+t$ then a client may have as side information,

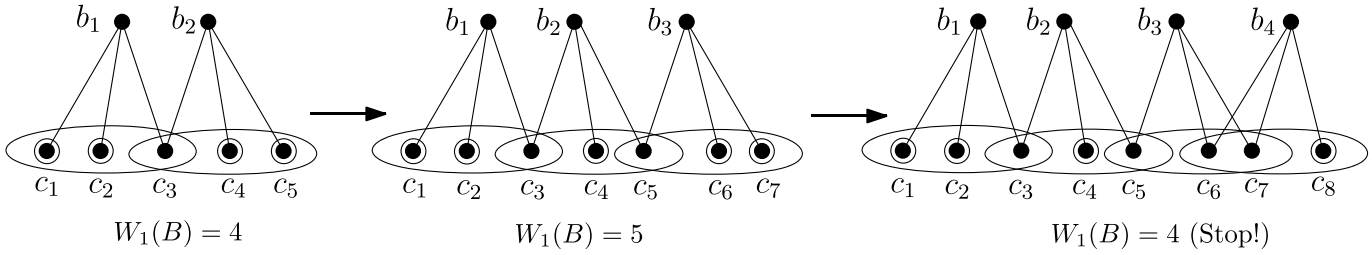

Fig. 4. Greedy construction of B with maximal $|W_1(B)|$.

messages corresponding to a subset of size s of U. But then, it cannot decode t new messages that it does not have as U was assumed to contain *all* the decoded message indices.

For simplicity, assume that $|U| = s + t$, $s + t \leq m - s$ and without loss of generality $U = \{1, 2, \ldots, s + t\}$. Let $A_\perp$ be the projection of A onto the first $s + t$ coordinates, i.e., the ones corresponding to U. Consider the following possible side information set:

$$S_1 = \{\mathbf{e}_{\alpha_{1,1}}, \ldots, \mathbf{e}_{\alpha_{1,s}}\} \tag{34}$$

such that $\alpha_{1,1}, \ldots, \alpha_{1,s} \notin U$ (such a set exists as $s + t \leq m - s$). Clearly, by the condition of decodability of a new message (Eq. (33)), at least one vector in e_U should be in $Span(A \cup S_1)$. Let it be $\mathbf{e}_{\gamma_1}$. Since the $\alpha_{1,i}$ indices are disjoint from U, $\mathbf{e}_{\gamma_1} \in Span(A_\perp)$. Now consider the side information set

$$S_2 = \{\mathbf{e}_{\gamma_1}, \mathbf{e}_{\alpha_{2,1}}, \ldots, \mathbf{e}_{\alpha_{2,s-1}}\} \tag{35}$$

where $\alpha_{2,1}, \ldots, \alpha_{2,s-1} \notin U$. In this case, an application of Eq. (33) implies that there exists $\mathbf{e}_{\gamma_2} \in Span(A \cup S_2)$. Since $\mathbf{e}_{\gamma_2}$ is orthogonal to all the vectors in S_2, $\mathbf{e}_{\gamma_2} \in Span(A_\perp)$. We can repeat this argument for a total of $s + 1$ steps where the last side information set is

$$S_{s+1} = \{\mathbf{e}_{\gamma_1}, \ldots, \mathbf{e}_{\gamma_s}\} \tag{36}$$

In this case, since t new messages need to be decoded by the client, we can conclude that there are vectors $\mathbf{e}_{\gamma_{s+1}}, \mathbf{e}_{\gamma_{s+2}}, \ldots, \mathbf{e}_{\gamma_{s+t}}$ that are not in S_{s+1} but are in $Span(A \cup S_{s+1})$, which implies they lie in $Span(A_\perp)$. Since the vectors $\mathbf{e}_{\gamma_1}, \ldots, \mathbf{e}_{\gamma_{s+t}}$ are orthogonal and all of them lie in $Span(A_\perp)$, $A_\perp$ and by implication A, must contain at least $s + t$ linearly independent vectors.

The other case where $s + t > m - s$ can be handled in a similar manner, where instead of $s + t$ sets, we have $m - s$ side information sets for which the new messages correspond to orthogonal vectors. By combining the two, we conclude that A must contain at least $\min\{s+t, m-s\}$ linearly independent vectors, which implies at least the same number of broadcasts need to be made. Finally, the assumption of $|U| = s + t$ can be removed by simple choosing the first $s + t$ elements of U.

The above argument can be repeated for each $s \in \Sigma$ and hence we obtain the tightest lower bound by taking the maximum of $\min\{s + t, m - s\}$ over all $s \in \Sigma$. ∎

Clearly, when $s_{\max} = s_{\min} = s$, there is only one element in Σ and the lower bound becomes $\min\{s + t, m - s\}$, which matches the upper bound in Lemma 10. This proves our claim in Thm. 9.

VII. Approximation Algorithms

In this section we propose polynomial time heuristic approximation algorithms for solving the PICOD(t) problem. Our main algorithm uses a greedy approach to iteratively find large subsets of clients that can be satisfied with a single coded message. In addition, we present two other algorithms specifically for PICOD(1) – one based on the upper bound proof in Section V and the other based on a reduction to the index coding problem.

A. Algorithm GRCOV

In this algorithm, using the graphical representation defined in Section V, we try to find a set of message vertices B such that $|W_1(B)|$ is maximized. Rather than trying to obtain the *maximum* such set, we greedily find a *maximal* such set. Let $B = \{b_{v_1}, \cdots, b_{v_t}\}$ be a set of message vertices. B is a maximal set if for any message vertex $b_{v_{t+1}} \notin B$, $|W_1(B \cup \{b_{v_{t+1}}\})| < |W_1(B)|$. To find a maximal set, we start with the empty set and keep on adding message vertices that greedily maximizes $|W_1(B)|$ in each step; we stop when no further additions are possible without decreasing $|W_1(B)|$. For example, Fig. 4 represents a possible sequence of operations where $B = \{b_1, b_2, b_3\}$ is a maximal set. When b_3 is added, the cardinality of $W_1(B)$ increases but further addition of b_4 decreases it, in which case we stop.

We maintain a counter $CNT[i]$ for each client vertex i to keep track of the remaining number of messages they request. For an instance of PICOD(t), they are all initialized to t. After finding a particular B for which $|W_1(B)|$ is maximal, we can use the sum of the messages in B to satisfy the message vertices in $W_1(B)$. The edges connecting the vertices in $W_1(B)$ to the message vertices it can decode are removed and the value of $CNT[i]$ for each vertex in $W_1(B)$ is also reduced by one. The algorithm is resumed for the remaining graph, until all the $CNT[i]$ values go to zero. We call this algorithm GRCOV (for greedy cover). It is shown in pseudo-code format below and a simple implementation of the algorithm has a running time of $O(mn^2t)$.

While we do not present worst case approximation guarantees for GRCOV in this paper, there exist examples where GRCOV can take $\Omega(\log n)$ messages while the optimal code

Algorithm 1 GRCOV(G, m, n, t)

Init: G is an instance of PICOD(t) with n client vertices and m message vertices. $\mathcal{C} = \{\}$, $CNT[i] = t$ for $i \in [n]$.
while $\exists i$ s.t. $CNT[i] \neq 0$ **do**
  $B \leftarrow \varnothing$.
  while B is not a *maximal set* **do**
    Find message vertex $b_v \notin B$ such that $|W_1(B \cup \{b_v\})|$ is maximized.
    $B \leftarrow B \cup \{b_v\}$.
  end while
  $\mathcal{C} \leftarrow \mathcal{C} \cup \left\{ \sum_{u=1}^{|B|} b_{v_u}, b_{v_u} \in B \right\}$.
  for $c_i \in W_1(B)$ **do**
    If c_i is connected to $b_j \in B$, then delete the corresponding edge in G.
    $CNT[i] \leftarrow CNT[i] - 1$.
  end for
end while
Output $\mathcal{C}$.

requires just two messages. We conjecture that GRCOV produces codes of size that are at most $O(\log n)$ times the optimal linear code.

B. Algorithm RANDCOV

The RANDCOV (for randomized cover) algorithm for PICOD(1) follows the procedure in the proof of Theorem 4 (in Section V) to find an encoding. The client vertices are partitioned into at most $g = O(\log m)$ non-empty groups $S_1, \ldots, S_g$ such that the ratio of the maximum and minimum degrees in each group is at most r (a fixed constant). Let the maximum degree of client vertices in S_i be $d_{\max,i}$. In the neighborhood $N[S_i]$, we select each vertex with probability $p_i = \frac{1}{d_{\max,i}}$. If B_i is the set of selected vertices, the clients in $W_1(B_i)$ are satisfied. These client vertices are removed and the process is continued until all the vertices in S_i are satisfied. The number of randomly sampled sets required is also the number of coded messages required. This is done for each of the sets S_i to find a code for all the client vertices. A simple implementation of the algorithm has an expected running time of $O(mn \log n)$.

1) Algorithm RANDCOV-PP Although the expected size of the code produced by RANDCOV is upper bounded by Thm. 4, as we shall see in the next section, a simple implementation does not perform very well as compared to GRCOV on random instances of PICOD(1). To make it more efficient we propose the following post processing phase. Let B_1 and B_2 be two sets of message vertices. If B_1 has no edges to $W_1(B_2)$ and B_2 has no edges to $W_1(B_1)$, we can send the sum of all the messages in $B_1 \cup B_2$ to satisfy all the client vertices in $W_1(B_1) \cup W_1(B_2)$. This can be extended to include more than two sets by selecting the sets greedily in order to minimize the number of coded messages. When this post-processing step is added to RANDCOV, we call the algorithm RANDCOV-PP. The expected running time of RANDCOV-PP remains $O(mn \log n)$.

C. Algorithm ICOD-SETCOV

Finally, we propose another algorithm for PICOD(1) that is based on a reduction to the index coding problem. In an instance of PICOD(1), it is sufficient that c_i is able to decode any single message in $N[c_i]$. To convert it into a form closer to an instance of the standard index coding problem, we split each client c_i into $|N[c_i]|$ "pseudo-clients" each, with a *distinct* message from $N[c_i]$ as a requirement and with the *same common* side information sets. Therefore, in total we have $\sum_{i=1}^{n} |N[c_i]|$ pseudo-clients. This is an instance of the index coding problem, which can be solved (non-optimally) using a clique-cover based greedy algorithm [3].

The greedy algorithm in fact produces a code $\mathcal{C}$ where each coded message is simply a sum of a set of messages. Each coded message $f_k \in \mathcal{C}$ satisfies a set of pseudo-clients which in turn correspond to the original clients. This naturally defines a covering relationship between the messages f_k and the set of clients, which we denote by $Cov(f_k)$ as shown below.

$$Cov(f_k) = \{c_{k,1}, \cdots, c_{k,s_k}\} \subseteq \{c_1, \cdots, c_n\} \tag{37}$$

In fact, because there can be multiple pseudo-clients corresponding to a client, the client can occur in several of these covering sets. Since we only need a client to be able to decode a single message that it does not know, it is sufficient to find a collection of f_k-s such that the corresponding $Cov(f_k)$-s cover all the clients. Further we want to minimize the size of this collection for the optimal encoding. This is precisely an instance of the minimum set cover problem with clients being the elements and the $Cov(f_k)$-s being the sets. In our implementation, we use the standard greedy approximation algorithm to solve it. Overall, the algorithm works by reducing an instance of PICOD(1) to an instance of index coding by splitting each client into several pseudo-clients and then solving a set cover problem defined on the index coding solution to minimize the number of encoded messages. We call this algorithm ICOD-SETCOV (for index coding set-cover) and the running time for a non-optimized implementation is $O(m^2 n^2)$.

VIII. Simulation Results

In this section, we present results of extensive simulations on random instances of PICOD(t) to evaluate the performance of the algorithms presented in Section VII. We first analyze the performance of all the algorithms for PICOD(1), namely GRCOV, RANDCOV, RANDCOV-PP and ICOD-SETCOV. We set both the number of clients (n) and number of messages (m) to be 512 and the field in which the messages reside is $\mathcal{F} = GF(2)$. For RANDCOV and RANDCOV-PP we choose $r = 3$. The random instances of PICOD(1) are generated as a function of the probability p_{msg} of a client having a particular message in its side information set. Thus, for each client c_i, a message m_j is in its side information set independently with probability p_{msg}. Equivalently, in the graph representation of the instance each edge is present with a probability of $1 - p_{\text{msg}}$. Such random instances can model block-fading in wireless channels, i.e., when the channel SNR is low, the higher network layers in the client experience erasures with probability $1 - p_{\text{msg}}$, while

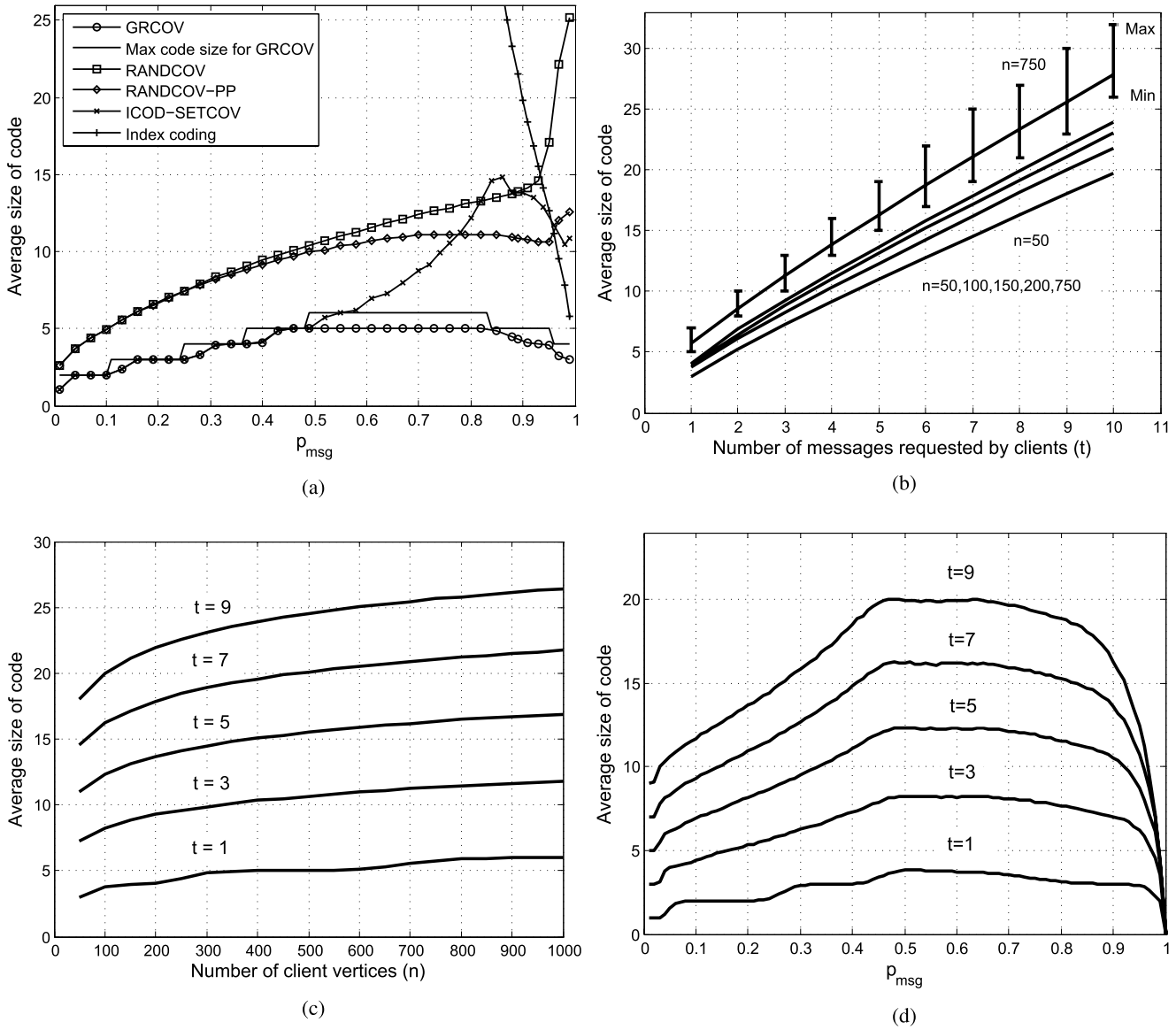

Fig. 5. Performance of algorithms for PICOD(1) and PICOD(t). (a) Average code size of PICOD(1) algorithms for varying p_{msg}. Here $m = n = 512$. (b) Average code size of GRCOV as a function of t as contours over n. Here $m = n$ and $p_{\text{msg}} = 0.5$. (c) Average code size of GRCOV as a function of n as contours over t. Here $m = n$ and $p_{\text{msg}} = 0.5$. (d) Average code size of GRCOV for varying p_{msg} and t. Here $m = n = 100$.

at a next block of high SNR, we want to perform "lossless" transmissions as efficiently as possible.

Fig. 5a shows the average performance of the algorithms over several runs. For each value of p_{msg}, we generate more than 10,000 instances and plot the average size of the codes produced by the different algorithms. For comparison, we also plot the average performance of the clique-cover based greedy algorithm presented in [3] for instances of the index coding problem generated with the same p_{msg}.

As expected, we observe a significant difference between the performance of the PICOD(1) algorithms and the index coding algorithm for the same p_{msg}. While all the three PICOD(1) algorithms produce codes of size less than 26 on average, the index coding solution hovers in this range only for $p_{\text{msg}} \geq 0.87$, which is the case when the side information sets are dense. In the remaining range of p_{msg} values, all the PICOD(1) algorithms use fewer messages and the difference only becomes larger when the side information sets are sparser.

Among the algorithms for PICOD(1) presented in the paper, GRCOV performs the best. For the random instances on which the simulations were run, arguments similar to the ones used in Section V can be used to show that it produces an encoding with the same asymptotic performance as RANDCOV, but the practical performance is much better. In fact, the maximum number of coded messages required by GRCOV (among the random instances in the simulation), which is also plotted in the figure, is not substantially different from the average number. The performance of RANDCOV-PP is substantially better than RANDCOV, especially when the side information sets are denser and hence G is sparser. Also, the performance of RANDCOV takes a hit in this regime. Both of these are due to the fact that the number of partitions in the client

vertices is relatively large, although most of them are "disjoint" which allows RANDCOV-PP to improve the performance significantly. Finally, ICOD-SETCOV performs as good as GRCOV when $p_{\text{msg}} \leq 0.5$ but becomes worse as G becomes sparser. This can be partly explained by the suboptimal nature of the greedy set-cover algorithm that we use as a part of ICOD-SETCOV.

To show the performance of the GRCOV algorithm for general PICOD(t), we generate random instances for a fixed $p_{\text{msg}} = 0.5$ and varying values of n and t. The number of messages m is taken to be equal to the number of clients n. If in an instance, the size of the side information set for a particular client vertex c_i is s and $m - s < t$, then $CNT[i]$ is initialized to $m - s$. We then plot the average size of the codes obtained versus t, for several values of n. The plot is shown in Fig. 5b.

The figure has several trends that shows the good performance of our algorithm. Notice that for each n, after a short initial phase, the average number of messages required grows almost linearly with t. This matches the trend expected from Theorem 8. Further, the contours representing different values of n in Fig. 5b are approximately parallel for $t > 5$, i.e., the slopes are independent of n. This also matches the trend suggested by an $O(t + \log^2 n)$ bound for large enough t. The error bars at the top, that represents the maximum (and minimum) size of the code encountered for random instances corresponding to $n = 750$, also shows an approximately linear trend.

In Fig. 5c, the parameter on the x-axis is changed to n and the lines correspond to different values of t. Finally, in Fig. 5, which is similar to Fig. 5a, we plot the dependence of the average code size with respect to p_{msg}, for different values of t and $m = n = 100$. In both figures, we see that the behavior does not change substantially for different values of t, with curves essentially getting translated upwards for higher values of t.

IX. Conclusion and Open Problems

In this paper we formulate the pliable index coding problem PICOD(t), where the clients are pliable and are interested in receiving any t messages that they do not have. Despite being NP-Hard to find the optimal pliable index code, we prove an upper bound on the number of broadcast messages required to satisfy all the clients that scales as $O(\min\{t \log^2 n, t \log n + \log^3 n\})$, for $m = O(n^\delta)$ messages and n clients (where $\delta > 0$ is a constant). When the side information sets are of equal cardinality, the size of the code scales as $O(\min\{t \log n, t + \log^2 n\})$. For the directly comparable case of $t = 1$, this is an exponential improvement over the worst case behavior in index coding. We develop efficient heuristic approximation algorithms for PICOD(t), that are shown to perform well on random instances. We also formulate the oblivious pliable index coding problem OB-PICOD(t) to model scenarios where the source only knows the cardinalities of the side information sets of the clients. We prove (constructive) upper bounds on the size of codes, that are shown to be tight for linear codes when the cardinalities of the side information sets are equal.

There are several directions for possible future research. In contrast to index coding, it is not trivial to construct instances of PICOD(t) that have provably large codes. Even if we restrict to the case of $t = 1$ and scalar (in addition to linear) codes, showing that PICOD(1) needs codes greater than a constant size is an open problem. In general, the question of the tightness of the upper bounds we prove, especially when t is a constant, remains open. In this paper, we only considered linear codes for PICOD(t). As in the case of index coding, where non-linear codes can be significantly smaller than linear ones for some instances, exploring the utility of such codes for PICOD(t) is a promising direction of future research. For OB-PICOD(t), since it is impossible to construct linear codes smaller than the ones obtained by our simple construction in many cases, designing smaller non-linear codes is a challenging open problem.

It is possible to take a more unified view of PICOD(t), OB-PICOD(t) and index coding. PICOD(t) and OB-PICOD(t) can be unified and generalized into a single problem by considering a scenario where the server only knows a subset of the side information sets of each client, in addition to their cardinalities. Index coding and PICOD can be unified by assigning individual weights to the messages for each client. Proving bounds on the size of codes for these unified versions will greatly generalize the results presented in this paper. Finally, we presented the GRCOV algorithm and showed that it has excellent performance on random instances of PICOD(t). Proving theoretical approximation guarantees on its performance is also a promising direction of future research.

Acknowledgment

The authors would like to thank the anonymous referees for their detailed comments and suggestions that have helped in improving the paper.

References

[1] S. Brahma and C. Fragouli, "Pliable index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 2251–2255.
[2] S. Brahma and C. Fragouli, "Pliable index coding: The multiple requests case," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 1142–1146.
[3] Y. Birk and T. Kol, "Informed-source coding-on-demand (ISCOD) over broadcast channels," in *Proc. IEEE 17th INFOCOM*, Mar./Apr. 1998, pp. 1257–1264.
[4] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1479–1494, Mar. 2011.
[5] R. Peeters, "Orthogonal representations over finite fields and the chromatic number of graphs," *Combinatorica*, vol. 16, no. 3, pp. 417–431, 1996.
[6] M. Langberg and A. Sprintson, "On the hardness of approximating the network coding capacity," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1008–1014, Feb. 2011.
[7] I. Haviv and M. Langberg, "On linear index coding for random graphs," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 2231–2235.
[8] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 2, pp. 24–36, 1997.
[9] E. Lubetzky and U. Stav, "Nonlinear index coding outperforming the linear optimum," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3544–3551, Aug. 2009.
[10] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3187–3195, Jul. 2010.

[11] A. Blasiak, R. Kleinberg, and E. Lubetzky. (2010). "Index coding via linear programming." [Online]. Available: http://arxiv.org/abs/1004.1379
[12] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Lexicographic products and the power of non-linear network coding," in *Proc. IEEE 52nd Annu. Symp. Found. Comput. Sci.*, Oct. 2011, pp. 609–618.
[13] H. Maleki, V. R. Cadambe, and S. A. Jafar, "Index coding—An interference alignment perspective," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5402–5432, Sep. 2014.
[14] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, "On the complementary index coding problem," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul./Aug. 2011, pp. 244–248.
[15] S. H. Dau, V. Skachek, and Y. M. Chee, "Optimal index codes with near-extreme rates," *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1515–1527, Mar. 2014.
[16] S. H. Dau, V. Skachek, and Y. M. Chee, "On the security of index coding with side information," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3975–3988, Jun. 2012.
[17] S. H. Dau, V. Skachek, and Y. M. Chee, "Error correction for index coding with side information," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1517–1531, Mar. 2013.
[18] Y. Berliner and M. Langberg, "Index coding with outerplanar side information," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul./Aug. 2011, pp. 806–810.
[19] A. S. Tehrani, A. G. Dimakis, and M. J. Neely, "Bipartite index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 2246–2250.
[20] M. J. Neely, A. S. Tehrani, and Z. Zhang, "Dynamic index coding for wireless broadcast networks," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7525–7540, Nov. 2013.
[21] L. Ong and C. K. Ho, "Optimal index codes for a class of multicast networks with receiver side information," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2012, pp. 2213–2218.
[22] A. Le, A. S. Tehrani, A. G. Dimakis, and A. Markopoulou, "Instantly decodable network codes for real-time applications," in *Proc. IEEE Int. Symp. Netw. Coding*, Jun. 2013, pp. 1–6.
[23] J. Blömer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman, "An XOR-based erasure-resilient coding scheme," Dept. Elect. Eng. Comput. Sci., Univ. Berkeley, Berkeley, CA, USA, Tech. Rep. TR-95-048, 1995.
[24] M. Shokrollahi, D. Spielman, and V. Stemann, "A remark on matrix rigidity," *Inf. Process. Lett.*, vol. 64, no. 6, pp. 283–285, 1997.
[25] T. J. Schaefer, "The complexity of satisfiability problems," in *Proc. 10th Annu. ACM Symp. Theory Comput.*, 1978, pp. 216–226.
[26] N. Alon and J. H. Spencer, *The Probabilistic Method*. New York, NY, USA: Wiley, 2008.
[27] D. P. Dubhashi and A. Panconesi, *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

Siddhartha Brahma received the B.Tech degree from the Indian Institute of Technology, Kharagpur in 2005, M.A. from Princeton University, USA in 2008 and Ph.D. from EPFL, Switzerland in 2015, all in computer science. His research interests are broadly in the area of algorithms, especially in wireless networks, network coding, machine learning and big data. He is the recipient of the best paper award at ACM MobiHoc in 2013 and the President of India gold medal at IIT Kharagpur in 2005. He is presently a Postdoctoral researcher at the University of Neuchâtel, Switzerland.

Christina Fragouli is a Professor at UCLA in the Electrical Engineering Department. She received the B.S. degree in Electrical Engineering from the National Technical University of Athens, Athens, Greece, in 1996, and the M.Sc. and Ph.D. degrees in Electrical Engineering from the University of California, Los Angeles, in 1998 and 2000, respectively. She has worked at the Information Sciences Center, AT&T Labs, Florham Park New Jersey, and the National University of Athens. She also visited Bell Laboratories, Murray Hill, NJ, and DIMACS, Rutgers University. Between 2006–2007, 2007–2012 and 2012–2015 she was an FNS Assistant Professor, an Assistant Professor and an Associate Professor, respectively, in the School of Computer and Communication Sciences, EPFL, Switzerland. She received the Fulbright Fellowship for her graduate studies, the Outstanding Ph.D. Student Award 2000-2001, UCLA, Electrical Engineering Department, the Zonta award 2008 in Switzerland, the Starting Investigator ERC award in 2009, the Mobihoc 2013 best paper award, and the MASCOTS 2011 best paper award. She served as an Associate Editor for IEEE COMMUNICATIONS LETTERS, for Elsevier Computer Communication, for IEEE TRANSACTIONS ON COMMUNICATIONS, for IEEE TRANSACTIONS ON INFORMATION THEORY, and for IEEE TRANSACTIONS ON MOBILE COMMUNICATIONS. Her research interests are in network coding, wireless networks, network security, and algorithms for networking.