Wireless Index Coding Through Rank Minimization

Jonathan I. Tamir*, Ethan R. Elenberg†, Anurag Banerjee†, and Sriram Vishwanath†

*Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California 94720

Email: jtamir@eecs.berkeley.edu

†Department of Electrical and Computer Engineering, University of Texas, Austin, Texas 78712

Email: {elenberg, anurag.banerjee}@utexas.edu, sriram@ece.utexas.edu

***Abstract*—Index coding, initially introduced within theoretical computer science to address a specialized class of problems, has gained significant interest within communications and networking communities in recent years. Index coding has been shown to be analogous to a large class of challenging wired network coding and wireless multi-terminal problems, the latter class being of primary interest in this paper. Here, a (relaxed) rank minimization based analytic framework is presented for wireless index coding, which represents a first step in a systematic algorithmic approach to index coding for practical use. Further, the paper demonstrates its applicability over a real-world wireless testbed. The scheme operates at the network layer, and can be understood as a (non-trivial) generalization of existing principles of random linear network coding. Experimental results demonstrate that, for a class of network topologies, the rank-minimized index coding system presents a throughput gain of 50 to 100 percent greater than random linear coding for this system.**

***Index Terms*—Downlink Communications, Wireless Protocol Design, Network Coding**

I. Introduction

Index coding forms a fairly well-studied body of research across multiple communities. Initially formulated as determining the throughput of a wired bottleneck network [1], there are multiple sources and destinations, none of which are directly connected to each other, and are instead connected through a bottleneck link. However, each destination is implicitly connected to some sources, and may use this *side information* in a decoding strategy. The goal of index coding is to derive structured coding strategies that will maximize the sum capacity of the entire network. As illustrated in [2], structured linear coding strategies can outperform time-division multiplexing (TDM) schemes, and determining the optimal coding strategy for the bottleneck link is referred to as the *Index Coding* problem.

Although specific index coding problems can be solved using techniques such as exhaustive search or interference alignment [2], the general network topologies problem is open. Indeed, index coding includes as its special cases, the general multiple unicast problem [3], [4] as well as a general $k \geq 3$ satisfiability problem [5], meaning that it is not only an open problem, but even when a solution is known to exist, finding it may be NP-hard. Thus, there is a growing literature on identifying classes of index coding problems where exact or approximate solutions can be found. [2], [6].

Index coding has recently found applications over wireless systems [2], [5], anycasting over networks as well as in efficient data storage. Specifically, in [6], the authors show a strong connection between the index coding problem and the multiple unicast problem with network coding. More recently, authors in [2] demonstrate the direct link between index coding and interference alignment, providing constructive examples where the concepts of interference alignment can be used to facilitate index coding.

In a relatively parallel line of work, the use of coding for wireless bottleneck networks has been studied, particularly for the case of interchange and downlink transmissions. In [7] among many other related papers, the authors demonstrate the benefits of random network coding (RNC) in enhancing the throughput of wireless bottleneck networks. In this existing body of work, network coding is performed in terms of choosing random coefficients for linear combinations of packets that are subsequently transmitted. The primary contribution of our work is to understand the benefits of choosing *structured* and *not* random combinations of packets using the index coding with interference alignment perspective. We refer to such a strategy as wireless index coding.

In this paper, our goal is to achieve efficient communication over a multi-way interchange wireless network setting, as illustrated by Figure 1. This setup consists of one access point/base station (AP) connected to multiple terminals wirelessly, both on the uplink (to the AP from the terminals) and downlink (to the terminals from the AP). Traditionally, for such a scenario, the AP may be understood as performing a round-robin based TDM in order to ensure that the right message is received by the appropriate destination. For example, Figure 1 shows the result of one such round, in which messages are received at some nodes (indicated by a ✓) and erased at others (indicated by an ×). Such a time-division scheme has been shown to be suboptimal for the two-party and multi-party interchange channel by many authors. Indeed, coded mechanisms for a multi-way exchange are known to minimize the number of transmissions required for the exchange channel and thus enhance throughput.

However, a static, pre-fixed coded strategy for the exchange channel can be inefficient in channels faced with errors and erasures. When transmissions may be corrupted and therefore, erased, it can prove beneficial to derive an *incremental* structure that exploits knowledge of those packets that are successfully received by destinations within the network. This concept corresponds to solving the wireless index coding problem for this exchange network, and we explain the problem formulation, its solution, and resulting throughput benefits in greater detail in subsequent sections of the paper.

In summary, the key contributions of this paper are as follows:

1) We formulate efficient communication over a multi-way interchange network as a wireless index coding problem, combining linear coding of packets with interference alignment.
2) We implement our solution over a hardware testbed and show a substantial benefit is obtained over both TDM and RNC by using structured coding.

II. System Model

Consider a wireless erasure network in which an AP broadcasts a set of messages $\mathcal{W} = \{W_1, \ldots, W_T\} \subseteq \mathbb{F}$ in discrete timeslots to K terminals. Each node R_k, $k = 1, \ldots, K$, desires a strict subset of the messages, $\mathcal{W}_{R_k} \subset \mathcal{W}$, has a probability ϵ_k of dropping a transmission, and has side information $\mathcal{A}_k \subseteq \mathcal{W} \setminus \mathcal{W}_{R_k}$. Throughout this paper, we assume $\mathcal{W}_{R_k} = \{W_k\}$ and $\mathbb{F} = \mathbb{R}$. Although it is common to formulate index coding over a finite alphabet, we have found that relaxing the problem to the reals, which is then truncated to a subset of the rationals for practical implementation, leads to algorithms that approximate the finite alphabet problem effectively, as evident by the experimental results in Section V.

In this setup, the AP gains knowledge of which transmissions are successful and/or erased within the network at the end of a transmission. In other words, both an acknowledgment (ACK) and a negative acknowledgment (NACK) are assumed for each case to be generated by the receivers and successfully received by the AP prior to its next transmission. We refer to this setup as a *multi-way interchange network*, since the entire goal is to perform a (generalized) exchange of data between nodes within the network. Such a model may be used to denote multiple categories of networks including sensor networks and relay-assisted multi-peer networks.

We represent such a network by the directed graph $\mathcal{G}(\mathcal{W}, \mathcal{A})$, where the vertex $k \equiv W_k \in \mathcal{W}$ represents the destination for message W_k and there is a directed edge from i to j if R_i possesses knowledge of the message the AP intends to send to R_j, i.e. if $W_j \in \mathcal{A}_i$. This model is also considered for wireline networks in [8].

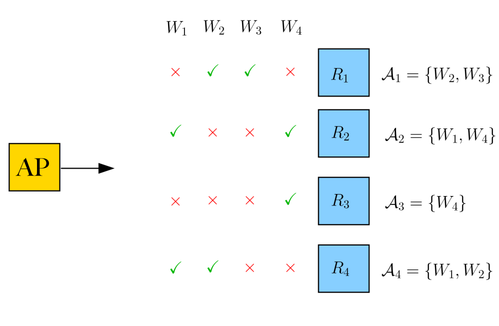


Fig. 1. AP broadcasts to $K = 4$ terminals over an erasure network using TDM. A ✓ indicates a received message and an × indicates an erasure.

III. Index Codes

To generalize the coding scheme to arbitrary networks, we begin with notation modified from [8]:

Definition 1 (Index Code [8]): An **index code** for $\mathcal{G}(\mathcal{W}, \mathcal{A})$ is given by

1) A set of codewords, $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$
2) An encoding function $E \colon \mathcal{W} \to \mathcal{S}$
3) Decoding functions $D_1, \ldots, D_K$ such that $D_k(E(\mathcal{W}), \mathcal{A}_k) = W_k$ for all $k \in \{1, \ldots, K\}$.

The length of the index code is $|\mathcal{S}| = m$.

Definition 2 (Index Coding Matrix [8]): The matrix $X = (x_{ij})$ is an **index coding matrix** for $\mathcal{G}(\mathcal{W}, \mathcal{A})$, denoted $X \in \mathcal{X}(\mathcal{W}, \mathcal{A})$, if

1) $x_{ii} \neq 0$
2) $x_{ij} = 0$ if $W_j \notin \mathcal{A}_i$

$\mathcal{X}$ is the space of all index coding matrices for $\mathcal{G}$.

Figure 2 shows the structure of an index coding matrix for the network in Figure 1 (after the initial TDM round) with received messages serving as side information for the next transmission. The numbers of X are fixed to satisfy the definition of an index coding matrix for $\mathcal{G}$, while the boxes can be arbitrarily chosen numbers.

$$X = \begin{array}{c} \\ R_1 \\ R_2 \\ R_3 \\ R_4 \end{array} \begin{array}{c} \begin{array}{cccc} W_1 & W_2 & W_3 & W_4 \end{array} \\ \begin{pmatrix} 1 & \square & \square & 0 \\ \square & 1 & 0 & \square \\ 0 & 0 & 1 & \square \\ \square & \square & 0 & 1 \end{pmatrix} \end{array}$$

Fig. 2. Index coding matrix for $\mathcal{G}$ in Figure 1. The numbers are fixed, while the boxes are arbitrary.

We see that any $X \in \mathcal{X}$ fully defines $\mathcal{G}$ — any linear combination of the form $X\mathbf{w}$, where $\mathbf{w} = \begin{bmatrix} W_1 & \cdots & W_K \end{bmatrix}^T$, will enable all nodes to decode their desired message. Row i in X represents the message coefficients observed by R_i. Because R_i desires W_i, it must have a non-zero coefficient in the i'th dimension. Similarly, if R_i does not have W_j as side information, it must have zero contribution in the j'th dimension. Any message available as side information has an arbitrary coefficient since it can be nulled out.

Theorem 1 ([8]): The optimum (shortest) linear index code for $\mathcal{G}$ is $m = \min_{X \in \mathcal{X}} \operatorname{rank} X$.

Since we can view the boxes in Figure 2 as missing entries, Theorem 1 establishes a direct connection to low-rank matrix completion theory [9], [10]. We wish to choose the arbitrary entries of X in a way that minimizes its rank, and thus minimizes the number of required transmissions. In a real-world environment, where packet losses are often correlated due to co-channel interference and fading [11], X is often low-rank. In this situation, multiple receivers share the same side information, which can lead to linearly dependent rows in the index coding matrix. For example, the boxes in Figure 2 can be chosen so that $\operatorname{rank} X = 2$.

The transmission strategy consists of an encoding scheme and a decoding scheme. For each $k \in \{1, \ldots, K\}$, define

$$\mathbf{a_k} := \begin{bmatrix} a_{k_1} & a_{k_2} & \cdots & a_{k_K} \end{bmatrix}^T, \tag{1}$$

where $a_{k_i} = W_i$ if $W_i \in \mathcal{A}_k$ and zero otherwise. Let $X \in \mathcal{X}$, and let $m = \operatorname{rank} X$. Since $X \in \mathbb{R}^{K\times K}$, $m \leq K$. Decompose X into two matrices, $V, U \in \mathbb{R}^{m\times K}$, so that

$$X = U^T V. \tag{2}$$

V represents a matrix of coefficients for m symbols, i.e. $E(\mathcal{W}) = V\mathbf{w}$. U is a zero-forcing matrix to ensure that $X \in \mathcal{X}$. Let $\mathbf{u_k}$ denote the k'th column vector of U. Then,

$$\mathbf{u_k}^T V\mathbf{w} - \mathbf{u_k}^T V\mathbf{a_k} = x_{kk} W_k \quad k = 1, \ldots, K. \tag{3}$$

By construction, W_k is uniquely determined up to a (known) constant. That is,

$$x_{kk} = \mathbf{u_k}^T \mathbf{v_k}, \tag{4}$$

where $\mathbf{v_k}$ is the k'th column vector of V.

We now use (1) – (3) and Theorem 1 to design our transmission scheme.

Transmission Strategy:

Encoding: Design V and U to satisfy (2), where $X = \underset{X' \in \mathcal{X}}{\operatorname{argmin}} \operatorname{rank} X'$.

Decoding: $D_k(E(\mathcal{W}), \mathcal{A}_k) \propto \mathbf{u_k}^T V\mathbf{w} - \mathbf{u_k}^T V\mathbf{a_k}$

This transmission strategy fully defines an index code for $\mathcal{G}$. The decoding scheme works for any V and U that satisfy (2). However, the AP requires knowledge of $\mathcal{A}$ for encoding and the terminals require knowledge of V for decoding. In practice, this requires sending the rows of V in the preambles of the symbols, resulting in some overhead, as well as sending ACK/NACK messages. Notice it is not necessary to send U, since it can be computed (if it exists) by

$$V_k \mathbf{u_k} = 0, \quad k = 1, \ldots, K, \tag{5}$$

where V_k is a submatrix of V corresponding to the concatenation of all column vectors that interfere with R_k. That is,

$$\mathbf{v_i} \in V_k \text{ if } W_i \notin \mathcal{A}_k \cup \{W_k\}, \quad i \in \{1, \ldots, K\}. \tag{6}$$

In addition to (5), we require

$$\mathbf{u_k}^T \mathbf{v_k} \neq 0 \tag{7}$$

so that W_k can be normalized by (4).

IV. Encoding Scheme and Transmission Algorithm

Section III establishes that the optimal transmission strategy fits the framework of low-rank matrix completion. This section describes an encoding scheme in this framework and the full transmission algorithm used in the implementation described in Section V.

We find a suitable X through low-rank matrix completion:

$$\mathcal{R}_1(m) := \begin{array}{ll} \underset{X' \in \mathbb{R}^{K\times K}}{\text{minimize}} & \left\| \mathcal{P}_\Omega(X') - b \right\|_2 \\ \text{subject to} & \operatorname{rank} X' \leq m \end{array}. \tag{8}$$

Here, $\mathcal{P}_\Omega \colon \mathbb{R}^{K\times K} \to \mathbb{R}^p$ represents the sampling operator [9], [10]. The set Ω corresponds to the indices that define an index coding matrix, and $b = \mathcal{P}_\Omega(\mathcal{X})$ specifies the defining constraints for any $X \in \mathcal{X}$. Let

$$X = \sum_{i=1}^{K} \sigma_i \tilde{\mathbf{u}}_\mathbf{i} \tilde{\mathbf{v}}_\mathbf{i}^T \tag{9}$$

be the singular value decomposition (SVD) of the solution to (8). Since the rank of X is m,

$$X = \sum_{i=1}^{m} \sigma_i \tilde{\mathbf{u}}_\mathbf{i} \tilde{\mathbf{v}}_\mathbf{i}^T \triangleq \tilde{U}\Sigma\tilde{V}^T. \tag{10}$$

Let

$$V = \tilde{V}^T, \tag{11}$$

$$U = (\tilde{U}\Sigma)^T. \tag{12}$$

Equations (11) and (12) satisfy (2).

Routine $\mathcal{R}_1$ is in general NP-hard, but it can be relaxed to a convex program [10] and solved using greedy algorithms. For this work we implemented $\mathcal{R}_1$ using "Atomic Decomposition for Minimum Rank Approximation" (ADMiRA) [9], which is in the family of matching pursuit methods. In each iteration, ADMiRA finds rank-m and rank-$2m$ matrix approximations via SVD, as well as solving a linear least-squares problem via conjugate gradient descent. The complexity is polynomial in the index code length.

Algorithm 1 shows the full transmission strategy for broadcasting the message set $\mathcal{W}$ to K nodes with side information $\mathcal{A}$. The first inner loop (lines 7–10) uses the routine $\mathcal{R}_1$ to choose m and design a rank-m index coding matrix with largest acceptable error (controlled by λ). The second inner loop (lines 13–17) transmits a single symbol at a time until either all m symbols are transmitted or a node indicates successful decoding, i.e.

$$\mathcal{W}_D := \{W_k : R_k \text{ decodes } W_k\} \neq \emptyset. \tag{13}$$

The outer loop repeats this process until all messages are decoded.

This strategy lends itself to a hardware implementation at the network layer for several reasons. First, the method takes advantage of existing ACK/NACK signals to update the system at the AP after each index coding iteration. Secondly, as in the case of RNC, a solution is guaranteed after each node receives K symbols. Thus, the worst-case encoding/decoding complexity depends only on the number of nodes and is equivalent to the case of RNC.

V. Hardware Implementation

To test the index coding transmission strategy, we designed and implemented a wireless testbed consisting of TinyOS-based sensor motes. We implemented the wireless sensor network downlink scenario, where nodes are paired and communicate through the AP, and we compared the throughput gains to those achieved using TDM and RNC. In this scenario, each node has its own message available as side information and the goal is to receive its corresponding pair's message.

Algorithm 1 Broadcast Index Coding Transmission Strategy

Input: $\mathcal{W}$, $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_K\}$, $\lambda > 0$

$t \leftarrow 0$
while $|\mathcal{W}| > 0$ **do**
  $t \leftarrow t + 1$
  $X^{(t)} \leftarrow \emptyset$
  $\mathcal{X} \leftarrow \mathcal{X}(\mathcal{W}, \mathcal{A})$
  $m \leftarrow 0$
  while $\left\| \mathcal{P}_\Omega \left(X^{(t)} \right) - \mathcal{P}_\Omega \left(\mathcal{X} \right) \right\|_2 > \lambda$ **do**
    $m \leftarrow m + 1$
    $\left(V^{(t)}, U^{(t)}, X^{(t)} \right) \leftarrow \mathcal{R}_1(m)$
  end while
  $\mathcal{S} \leftarrow V^{(t)}\mathbf{w}$
  $i \leftarrow 0$
  while $|\mathcal{W}_D| = 0$ and $i < m$ **do**
    $i \leftarrow i + 1$
    Transmit $\mathcal{S}_i$
    $\mathcal{W}_D \leftarrow$ decoded messages
  end while
  $\mathcal{W} \leftarrow \mathcal{W} \setminus \mathcal{W}_D$
  $K \leftarrow K - |\mathcal{W}_D|$
end while

A. Experimental Setup

The hardware setup is shown in Figure 3 and consists of the following three components:

1) AP Host: The AP Host is a desktop computer that uses the messages and side information to perform the first inner loop of Algorithm 1. Routine $\mathcal{R}_1$ was implemented in Python for quick prototyping and ease of interfacing with the TinyOS network stack. This by no means represents a real-time implementation, which can leverage dedicated processing and algorithms.

2) AP Target: After computing $\mathcal{S} = V\mathbf{w}$, the AP Host transmits a single symbol $\mathcal{S}_i$ to the AP Target through a direct Ethernet connection. The AP Target is a MICAz radio mote [12] attached to the MIB600 Ethernet Interface Board. The MIB600 enables direct Ethernet connection to the AP Host for both programming and collecting experimental results.

3) Terminal Mote: After the AP Target receives a symbol from the AP Host, it broadcasts the symbol to the Terminal Motes in the wireless network via the TinyOS protocol. The terminals, which are MICAz radio motes scattered around the AP Target at varying distances, attempt to decode their desired message after each broadcast symbol and respond to the AP Target via ACK/NACK signals. The terminals implement the decoding scheme (5) using SVD. Note that for a real-time, scalable implementation, the terminals could alternatively use Gaussian elimination similar to [13].

Given the motes' limited computational capabilities, we implement the computationally demanding greedy alignment encoding scheme (Section IV) on the AP Host. Although this does not represent a true real-time implementation, we argue that the testbed unequivocally demonstrates an index-coding setup which can be extended to a full real-time implementation with a single AP component. Further, recent advances in low-rank matrix completion, e.g. iterative soft-thresholding, have led to algorithms with reduced complexity [10].

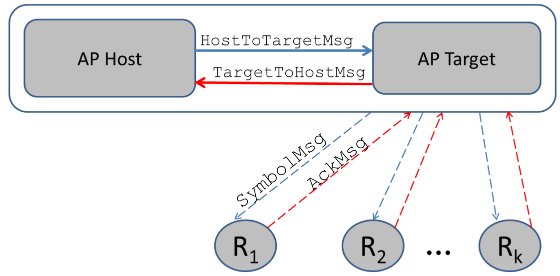


Fig. 3. Communication diagram.

TABLE 1
PACKET STRUCTURE

Packet/Field	Type	# of Bits	Description
`SymbolMsg` and `HostToTargetMsg`:			
`V_coeff`	`nx_float[K]`	$32 \times K$	Message coefficients
`data`	`nx_float[B]`	$32 \times B$	Symbol data
`messageid`	`nx_uint16`	16	TDM destination
`c_row`	`nx_uint16`	16	Synchronization
`AckMsg`:			
`nodeid`	`nx_uint16`	16	Responding node
`ack_type`	`nx_uint16`	16	Decoding status
`c_row`	`nx_uint16`	16	Synchronization
`TargetToHostMsg`:			
`ack_type`	`nx_uint16[K]`	$16 \times K$	All node ACKs
`c_row`	`nx_uint16`	16	Synchronization

We tested several spatial topologies throughout our experiments; however, the placement of the terminals generally formed a semicircular arc around the AP Target. The radius at which the terminals were placed varied so as to modify the erasure probabilities in the network. Operating statistics could then be collected across a broad range of erasure probabilities.

Prior to running the index coding part of the experiment, the AP Host generates random messages and distributes them to the corresponding terminals through a TDM scheme. For each successive round, new messages are generated which requires new side information to be sent to each terminal. Thus, the redistribution of side information to the terminals each round also enables computation of TDM metrics, such as the network erasures and number of transmissions.

Computationally, a number of matrix data structures must be initialized on the motes so that symbols sent by the AP Target can be used for decoding. When a mote receives its side information (indicating a new TDM round), it must reinitialize its data structures. Table 1 shows the data structures comprising the three packets communicated over the network. The AP Host sends to the AP Target a `HostToTargetMsg` packet, which contains the current symbol and the coefficients used to generate the symbol. The AP Target then broadcasts this packet to the nodes as a `SymbolMsg`. While coefficients

`V_coeff` and message symbols `data` are truncated from double to single precision floating-point values for transmission, we found that finite precision effects did not hinder the ability to decode messages at the receivers. Note that RNC is easily incorporated in this setup by randomly sampling `V_coeff` from a standard normal distribution. The current transmission/time slot is tracked to ensure that the motes are operating synchronously. The proportion of header information to raw data in the `SymbolMsg` packet is variable depending on the initial number of motes in the network, K, and the number of 32-bit words per message, B. We define the data efficiency, ϕ_B^K, as the proportion of raw data to total data:

$$\phi_B^K = \frac{B}{K + B + 2}. \tag{14}$$

For an eight-mote setup with four words per message, the efficiency is $\phi_4^8 = \frac{2}{7}$. Although this efficiency is low, a robust implementation will code over several kilobytes of data, making the header information negligible in size.

After each `SymbolMsg` transmission, the motes send an `AckMsg` packet to the AP Target. The ACK/NACK setup is designed such that a node sets `ack_type` to '1' if it received the symbol and was able to decode, and '0' if it received the symbol but was unable to decode. If the AP Target does not receive an `AckMsg` from a node, it defaults `ack_type` to '2' to indicate a NACK. Based on this setup, the erasures are two-sided.

The AP Target aggregates all ACKS and NACKs from the nodes and sends them to the AP Host in a `TargetToHostMsg` packet. The AP Host uses this information to estimate erasure probabilities at each node and update its list of nodes still in the network. After all nodes decode their message (and exit the system), the AP Host logs the transmission statistics and prepares the testbed for a new round by generating new messages and switching to TDM mode.

B. Experimental Results

We determine the performance of index coding over TDM as the ratio of number of transmissions at a particular measured erasure probability ϵ averaged over all K nodes. For a particular (ϵ, K) pair, define the index coding gain as

$$\eta_\epsilon^K := \frac{\text{\# of transmissions using TDM}}{\text{\# of transmissions using index coding}}, \tag{15}$$

and similarly define ν_ϵ^K as the corresponding RNC gain.

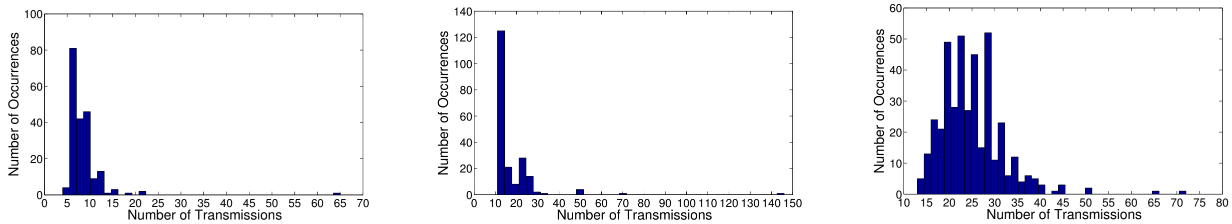
In our experiments, the network size K varied from 4 to 12 nodes while the transmission time slot was fixed at 2 seconds. Testing was conducted in an office setting with several reflective surfaces and RF interference sources. Several network topologies and distances were used to produce a range of erasure probabilities. Data size was adjusted by changing B, the number of 32-bit words per message, to attain reasonable values of ϵ as follows: $B = 1$ for $K = 12$, $B \in \{1, 4\}$ for $K = 8$, and $B \in \{4, 6\}$ for $K = 4$. Topologies included a semicircular setup with each node less than 0.5 m from the transmitter, a semicircular setup with nodes 2 m from the transmitter, placement throughout a room with transmitter distances of 0.5 m - 4 m, and a paired topology where paired nodes were placed on opposite ends of the transmitter at a distance of 2 m. While the data rates and packet sizes are scaled down, this setup is a reasonable step toward low-bandwidth sensor network applications.

Nearly 4000 rounds of experiments were conducted across all values of K, transmission schemes, and network topologies. The erasure probabilities for each node varied greatly from round to round and were often asymmetric during any given round. For large K, co-channel interference caused a reduced range of erasure probabilities irrespective of mote placement. For small K, erasures were much more varied because the link was more susceptible to ambient RF interference. Iterations of RNC and index coding were alternated after no more than 150 rounds to ensure consistency of the wireless channel. Coding gain was calculated by comparing each index coding round to the TDM transmission immediately preceding it.

Figure 4 shows histograms of the number of transmissions required to deliver all $K = 12$ messages using index coding, RNC, and TDM respectively. Note that both index coding and RNC have a similar spread; however index coding has a mean number of transmissions of about 9, whereas the mean for RNC is about 17. TDM has the largest average number of transmissions at 25, as well as the largest spread. Thus, index coding is the most reliable and consistent of the three schemes.

Figure 5 shows average coding gains for each scheme after the erasure probabilities were binned into intervals of 5%. Averages were calculated in a bin only if it contained at least 80 data points. Clearly, structured index coding performs better than RNC in the sensor network case (paired nodes). The largest average gain for RNC is $\nu_{.475}^{12} = 1.76$, while the largest average index coding gain is $\eta_{.425}^{12} = 3.70$. Note the highest gains occurred for both coding schemes in the 12-node setup. One can attribute this to the higher co-channel interference, thus causing correlated erasure probabilities. In addition, the larger number of nodes enables more opportunities for interference alignment as nodes exit the system. Incidentally, erasures were more variable in the case of index coding, which caused the curves to appear slightly less smooth than the random coding plots for the same number of trials.

Note that for high erasure probabilities the coding gain exceeds 2, the theoretical gain for $\epsilon = 0$. This is likely due to positive correlation among erasure probabilities for different nodes. If different nodes see similar channels, then they are more likely to receive the same symbols and exit the system at the same time, reducing the number of transmissions. In the case of ϵ_i independent and identically distributed for each node, RNC gain decreases as a function of ϵ and performs worse than TDM for high erasure probability [14]. Here, performance improves for both coding schemes, but index coding allows for greater exploitation of correlated erasures. Table 2 shows the gain obtained by comparing index coding to RNC at each erasure bin for each value of K. The largest coding gain of 1.97 occurs at $\epsilon = .35$ for $K = 4$. In this basic example, index coding has a theoretical maximum advantage



(a) Index Coding (b) RNC (c) TDM

Fig. 4. Number of transmissions for (a) index coding, (b) RNC, and (c) TDM at $K = 12$.

TABLE 2
CODING GAIN OVER RANDOM CODING: $(\eta_\epsilon^K / \nu_\epsilon^K)$

K	Erasure Bin (Percent)									
	$0-5$	$5-10$	$10-15$	$15-20$	$20-25$	$25-30$	$30-35$	$35-40$	$40-45$	$45-50$
4	1.49	1.49	1.45	1.47	1.58	1.53	1.97	-	-	-
8	-	-	-	-	1.55	1.57	1.53	1.70	-	-
12	-	-	-	-	-	-	-	1.78	1.72	-

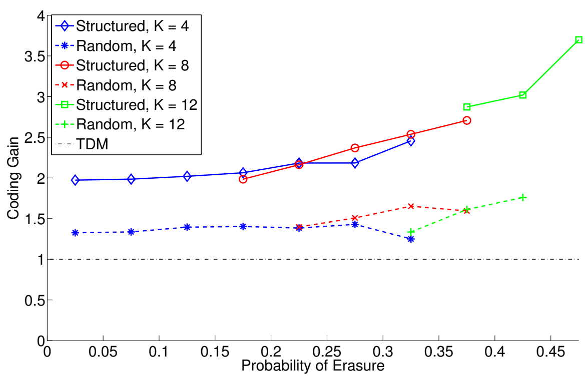


Fig. 5. Average Coding Gain Over TDM.

over RNC of

$$\frac{\eta_0^K}{\nu_0^K} = \frac{2(K-1)}{K}. \tag{16}$$

Experimental data confirms that this advantage is achievable in a real-world, high erasure setting.

VI. CONCLUSIONS

This paper presents a greedy rank minimization framework for wireless index coding and demonstrates its applicability to the multi-way interchange network with a non-trivial hardware implementation. Using structured transmission strategies, particularly in the case of correlated erasures, index coding can provide up to twice the throughput of RNC. Future research may include more complex network topologies, where differing side information at each node can be better exploited for throughput gains.

ACKNOWLEDGEMENTS

We would like to thank Atresh Sanne and Kannan Srinivasan for their help throughout this work.

REFERENCES

[1] Y. Birk and T. Kol, "Informed-source coding-on-demand (ISCOD) over broadcast channels," in *INFOCOM*, vol. 3. IEEE, 1998, pp. 1257–1264.

[2] H. Maleki, V. Cadambe, and S. Jafar, "Index coding: An interference alignment perspective," in *Proc. IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2012, pp. 2236–2240.

[3] M. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, "On the complementary index coding problem," in *Proc. IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2011, pp. 244–248.

[4] A. Das, S. Vishwanath, S. Jafar, and A. Markopoulou, "Network coding for multiple unicasts: An interference alignment approach," in *Proc. IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2010, pp. 1878–1882.

[5] M. Chaudhry and A. Sprintson, "Efficient algorithms for index coding," in *IEEE INFOCOM Workshops 2008*. IEEE, 2008, pp. 1–4.

[6] 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, 2010.

[7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 243–254, Aug. 2006.

[8] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1479–1494, 2011.

[9] K. Lee and Y. Bresler, "Efficient and guaranteed rank minimization by atomic decomposition," in *Proc. 2009 IEEE International Symposium on Information Theory - Volume 1*, May 2009, pp. 314–318.

[10] E. Candes and B. Recht, "Exact low-rank matrix completion via convex optimization," in *Communication, Control, and Computing, 46th Annual Allerton Conference on*, Sept. 2008, pp. 806 –812.

[11] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer," *IEEE Trans. Ind. Electron.*, vol. 49, no. 6, pp. 1265 – 1282, Dec 2002.

[12] *MICAz Wireless Measurement System Spec Sheet*. [Online]. Available: http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

[13] A. Hagedorn, D. Starobinski, and A. Trachtenberg, "Rateless deluge: Over-the-air programming of wireless sensor networks using random linear codes," in *Proc. 7th International Conference on Information Processing in Sensor Networks*, ser. IPSN '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 457–466.

[14] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 914 –925, Feb 2009.