Energy Minimization with Network Coding via Latin Hypercubes

Murat Kocaoglu, *Student Member, IEEE,* Ozgur B. Akan, *Senior Member, IEEE*

Abstract—**Network coding is mostly used to achieve the capacity of communication networks. In this study, motivated by the nanoscale communications where the energy cost for the channel symbols are asymmetric due to the widely employed OOK modulation, we design energy minimizing network codes. We develop the best mapping between the input and the output symbols at the network coding node that minimizes the average codeword energy using Latin squares, which we call the minimum energy network code (MENC). We define the class of networks composed of coding nodes with N incoming and 1 outgoing symbols as In-N networks. First, we derive the condition on the network code to minimize the average energy in In-Two networks and propose two linear MENCs. Later, we investigate the minimum energy network codes for In-N networks using the Latin hypercubes and propose a low energy network code (LENC) to reduce the average energy with network coding. We compare MENC with the classical XOR and random network codes for In-Two networks. The performance comparison between LENC and random network codes for In-N networks shows that the proposed network codes provide significant energy gains.**

Index Terms—**energy efficient network codes, green communications, minimum energy coding, network coding, latin squares.**

I. Introduction

Network coding is the method of combining the information flows at the relay node, which is essential to achieve the network capacity in certain networks, as routing only is not sufficient in general [1]. From the day it was first proposed, network coding has drawn great interest from the community. In addition to purely information theoretic analyses, researchers have incorporated network coding into various subjects, from cognitive radio to ad-hoc networks [4], [9] to vehicular networks [2], [3]. Even though the main research on network coding focuses on achieving the network capacity, or its gains in terms of the achievable rate [1]-[8], there are several papers discussing the energy efficiency aspects of network coding [9]-[14]. However, in most of these studies, network coding is not used directly as a tool to minimize the

M. Kocaoglu was with the Next-Generation and Wireless Communications Laboratory, Department of Electrical and Electronics Engineering, Koc University, Istanbul 34450, Turkey. He is now with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin TX 78712 USA (e-mail: mkocaoglu@ku.edu.tr)
O. B. Akan is with the Next-generation and Wireless Communications Laboratory, Department of Electrical and Electronics Engineering, Koc University, Istanbul 34450, Turkey (e-mail: akan@ku.edu.tr).

average energy per channel symbol, but energy efficiency is rather analyzed using the known network codes. If we can find network codes that minimize the average energy among all the input-output mappings in the network coding node, we can achieve capacity, i.e., maximize rate, with minimum energy dissipation in the network. Hence, using network coding to minimize energy dissipation is an important step in the characterization of networks, as it could lead to optimum networks operating at the best energy-rate pair.

In this paper, we propose such a network coding technique that minimizes the energy required at the transmitter. We show that network codes can be designed to minimize average energy, in addition to achieving the capacity in certain scenarios. We consider the scenario where the underlying channel symbols are asymmetric in the following way: One of the channel symbols consumes less energy than the others. Average energy per channel symbol can then be reduced by increasing the frequency of the modulation state with lower energy. The proposed technique exploits this idea to select the most energy-efficient mapping as the network code. Notice that the term "channel symbol" can refer to the modulated symbols or the codewords depending on the network assumptions.

Consider a relay node R with two incoming edges. We want to process the incoming channel symbols u_1 and u_2 such that the average energy per channel symbol is minimized at node R. Note that the output alphabet of the network code is assumed to be the same as the input alphabet, i.e., the same channel symbols are used throughout the network. We must give priority to the main goal of network coding during this task, i.e., achieving capacity. Notice that the notion of capacity that we are interested in is not on the physical channel level: We assume each symbol can be reliably transmitted over the physical channel. The capacity refers to the fact that the network code does not result in loss of information at the decoding node. Without knowing the rest of the network, we cannot determine the network code which guarantees achieving the network capacity. However, if we can *preserve information* with the network coding operation at R, sink nodes can decode the information that they desire using side information from other nodes. Such a mapping can then be used in general network topologies, since it preserves information independent from the network topology. We call such mappings, i.e., mappings which preserve information as *reversible* network codes. See Sec. III for a formal definition.

In this work, we propose a network code that exploits the asymmetry in the energy costs of channel symbols. We first develop the energy minimizing network code in Section IV, which we call *the minimum energy network code* (MENC), that

ensures the minimum energy dissipation in In-Two networks, i.e., networks composed of network coding nodes with two input and one output symbols, assuming asymmetrical energy cost of the channel symbol. This network code maps the incoming channel symbols to the outgoing symbols such that the average energy per output symbol is minimized. We then show the existence of *linear* minimum energy network codes. In Section V, we investigate the average energy at the transmitter, assuming minimum energy coding as the underlying channel code, which we have proposed in [16], [17] and used for nanoscale ad-hoc networks in [18]. In Section VI, we investigate the generalized In-N networks, i.e., networks composed of coding nodes with N incoming and one outgoing edges, and propose a novel network code to *reduce* energy, if not minimize, which we call *the low energy network code* (LENC). Compared to the random network codes. LENC can be used to achieve network capacity, while guaranteeing low energy dissipation. Performances of MENC and LENC are compared with XOR and random codes in Section VII.

II. RELATED WORK

Network coding is developed in [1], where authors show that routing only is not sufficient to achieve the network capacity and processing of the data flows at the intermediate nodes is required. Network coding idea has been employed in various settings since. Physical layer network coding is proposed in [19], where the authors exploit the broadcast nature of wireless nodes in ad-hoc networks to improve the throughput. The main research effort on network coding focuses on the capacity of networks. In this regard, authors in [5] show that linear coding is sufficient to achieve the network capacity in multicast networks. An algebraic framework for linear network codes is developed in [6]. In [7], the authors show that linear network coding is insufficient in general, even for large finite fields and with vector extensions, using a counter example.

There are several studies investigating the energy efficiency of network coding. In [9], a linear program that determines the path yielding the minimum energy per bit is developed in multicast networks with network coding. Each link is assumed to have an associated cost. It is shown that with network coding, energy per bit is reduced, and also the minimum energy solution can be found in polynomial time. However, authors do not design network codes to achieve energy efficiency, but rather exploit the idea of reducing the number of transmission with network coding. Authors in [10] optimize the network resources in coded wired and wireless networks to minimize a given cost criterion. In [11], energy efficient communication is achieved in wireless networks, by decomposing the network coding sessions into multicast and unicast. Moreover, optimization algorithms for solving the link scheduling problem are investigated. In [12], authors minimize the total energy consumed in the network with lifetime constraints, by determining the traffic on each edge, assuming XOR coding. In [13], an optimization problem with rate constraint is developed to minimize energy in wireless multi-hop networks. However, authors incorporate the routing and scheduling into the optimization, using the fixed XOR coding.

Therefore, energy minimization is not obtained with network coding. In [14], codes are designed in a P2P communication setup to reduce energy dissipation at the decoder. We, instead, design the network codes to minimize the energy dissipation at the transmitter node. This is especially important in the nanoscale communication scenarios [16] where the decoder can be a micronode with little to no energy concern, whereas the nanoscale transmitter has a very limited energy budget.

In this work, we design network codes with the objective of energy minimization when one of the channel symbols has a smaller energy cost. The network code minimizes the average energy per codeword at the outgoing edges of the relay node.

III. NETWORK MODEL

We mainly employ the notations adopted from [8]. A network is represented with a directed graph with edge set ε and node set ν. The finite set μ with cardinality M, i.e., $|\mu| = M$, is called the message set.

A set $\mathcal{A}$ with minimum of two elements is called an alphabet. The alphabet contains the channel symbols used at the edges of the network. The map $\mathcal{V} : \mu \to \mathcal{A}$ is the assignment of messages to the channel symbols. For unique decodability of source messages, $\mathcal{V}$ is a *bijection*, which is determined by the underlying channel code. Therefore, the size of the alphabet $\mathcal{A}$ is equal to the source set cardinality, i.e., $|\mathcal{A}| = |\mu| = M$. Without loss of generality, we take $\mathcal{A} = \{0, 1, ..., M - 1\}$ for simplicity.

Input and output edge sets of node u are shown by Γ_{in}^u and Γ_{out}^u, respectively. A node u is called a *source node* if Γ_{in}^u is empty, and a *destination node* if Γ_{out}^u is empty. The source and destination nodes are called the *end nodes* and the remaining nodes in the network are called the *relay nodes*. The set of channel symbols delivered to node u on its input edges is $In(u)$, and set of symbols generated by u is $Out(u)$. Let τ be the number of incoming edges of the relay node u, i.e., $|In(u)| = \tau$. Then, for each edge $e \in \Gamma_{out}^u$, the mapping

$$f_u^e(In(u)) : \mathcal{A}^\tau \to \mathcal{A}, \qquad (1)$$

is the network code at node u. We call the network code *reversible* if, given any $\tau - 1$ input messages and the message at any outgoing edge e of node u, i.e., f_u^e, the unknown message at the input can be uniquely determined. This condition imposes the *conservation of information at u*. We assume,

- for any relay node u, $|Out(u)| = 1$. Therefore, only a single channel symbol is transmitted from each node. The most common example of this scenario is in wireless networks, where a single channel symbol is broadcast due to the nature of the wireless channel. Hence, we say $f_u^e = f_u, \forall e \in \Gamma_{out}^u, \forall u \in \nu$.
- all the network coding nodes in the network employ *reversible* network coding.

It is important to note that there may be scenarios in which loss of information due to irreversible network codes can be compensated from other edges. However, in this work, in an attempt to develop codes where network coding is required to achieve the capacity, we assume reversible network codes.

IV. MENC FOR IN-TWO NETWORKS

In-Two networks are composed of coding nodes with two incoming and one outgoing symbols, i.e., for coding node u, $|In(u)| = 2$ and $|Out(u)| = 1$. Network may also consist of forwarding nodes, with $|In(u)| = |Out(u)| = 1$. In order to minimize average transmission energy at the network coding node, we need to consider the channel symbol probabilities. The probability distribution of the outgoing symbols at the network coding node depends on the probabilities of the incoming channel symbols. For a relay node u, $In(u) = \{u_1, u_2\}$ and $\Gamma_{in}^u = \{e_1, e_2\}$, where e_i contains the channel symbol u_i. We familiarize the reader with the *latin squares* next, which are useful to work on the reversible In-Two network codes.

A. Latin Squares as Network Codes

Latin squares are two dimensional mathematical objects useful to define reversible network codes.

Definition: A latin square of order M is an $M \times M$ square matrix containing each symbol from a set of cardinality M exactly once in each row and each column.

Due to their nature, latin squares can be used to represent reversible network codes for In-Two networks. Notice that latin squares are different from circulant matrices, since the former does not have restricted shift properties. In this work, the origin of a latin square is assumed to be its top-left corner. Assume that the $i + 1^{th}$ row (column) shows that the symbol i is received on the first (second) incoming edge of an In-Two node u, i.e., $u_1 = i$ ($u_2 = i$). For example, if we know that the first edge contains channel symbol 3, and the outgoing edge contains channel symbol 1, we can infer that the other incoming message should be 2, since each symbol appears exactly once in each row and each column. Similarly, the well-known XOR network code can be represented by a 2×2 latin square with zeros in the diagonal. It can be written as

$$f_u(u_1, u_2) = u_1 + u_2 \pmod{2}, \tag{2}$$

where channel symbols are selected from the binary field. Therefore when f_u and either of u_i's are known, the unknown input can be uniquely determined by $u_1 = f_u + u_2, \pmod{2}$.

B. Minimum Energy Network Coding - MENC

In this section we develop the energy minimizing network codes. As previously described, network codes map the incoming channel symbols to the outgoing channel symbols. We assume that the energy dissipation associated with each channel symbol is the same for all but smaller for only one of the symbols. Without loss of generality, we assume that channel symbol 0 is associated with energy dissipation of ψ_0, and all other symbols are associated with ψ_1, where $\psi_0 < \psi_1$.

We first present three lemmas which are useful for the proof of our main theorem. The following lemma gives the condition on the completability of partially filled latin squares.

Lemma 1. *[20] An $M \times M$ partially filled latin square with $M - 1$ entries can always be completed to a latin square.*

Lemma 1 states that if a square matrix filled with $M - 1$ entries satisfies the latin condition, then it can always be completed to a latin square. Using this, we immediately assert the following Lemma:

Lemma 2. *An $M \times M$ partially filled latin square with M symbols of the same type can always be completed to a latin square.*

Proof. Since latin square with $M - 1$ symbols of the same type can always be completed from Lemma 1, latin square with M symbols of the same type can also be completed, as in that case, the location for the Mth symbol is fixed due to the latin square condition. □

The average energy is determined by the positions of the 0 symbols in the latin square, since all the other symbols are associated with the same amount of energy dissipation. Therefore, we can work with latin squares filled with 0 symbol to find energy minimizing network codes. These partially filled latin squares always correspond to valid network codes, since they can always be completed by Lemma 2. To minimize average energy, we maximize the probability of the channel symbol associated with smaller energy dissipation.

Assume that the symbols are ordered in terms of their probabilities in the rows and columns of the latin square, i.e., i^{th} symbol on the first (second) incoming edge of node u is more probable than the jth symbol on the first (second) incoming edge, if $i < j$, i.e., $p_i \geq p_j$ and $q_i \geq q_j$ for $i < j$, where p_i and q_i stand for the corresponding probabilities.

Let us represent $x + 1$ by $\tilde{x}$ for $x \in \{i, j, m, n\}$. Hence, $\tilde{i}^{th}$ row corresponds to the incoming symbol with probability p_i. Without loss of generality, assume that $i < j$. We introduce the *interchange operation* as the switching of $\tilde{i}^{th}$ and $\tilde{j}^{th}$ rows of a latin square, where 0 symbol of the $\tilde{i}^{th}$ row is on the $\tilde{m}^{th}$ column and 0 symbol of the $\tilde{j}^{th}$ row is on the $\tilde{n}^{th}$ column. Then we have the following lemma.

Lemma 3. *The interchange operation decreases the average energy on the outgoing edge of network coding node, if $m > n$.*

Proof. We know that $\psi_0 < \psi_i, \forall i \in \mathcal{A}$ and $\psi_i = \psi_j = \psi_1, \forall i, j$ such that $i \neq 0, j \neq 0$. Let the average energy before the interchange operation be W_b and after the operation be W_a. Then we have,

$$\begin{aligned}
W_b - W_a &= \psi_0(p_i q_m + p_j q_n) + \psi_1(p_i q_n + p_j q_m) \\
&\quad - \psi_0(p_j q_m + p_i q_n) - \psi_1(p_i q_m + p_j q_n) \\
&= (\psi_1 - \psi_0)(p_i - p_j)(q_n - q_m) \tag{3}
\end{aligned}$$

In (3), the right hand side is always positive. Hence, the interchange operation decreases the average energy. □

Theorem 1. *A minimum energy network code for a node with two incoming links is given by a latin square with 0s on the main diagonal.*

Proof. As stated previously, the average energy per channel symbol depends on the positions of the 0 symbols on the latin square representing the network code, since all the other channel symbols has the same energy cost.

Due to the 2-dimensional nature of latin squares, starting from any partially filled latin square of order M filled with M

number of 0s, we can obtain all such latin squares by applying sufficient number of row or column switching operations.

From Lemma 3 we know that, to minimize the average energy, we should apply row switching operations until the index of the column containing the 0 symbol for $\tilde{i}^{th}$ row is less than that of $\tilde{j}^{th}$ row for all i, j satisfying $i < j$. Sufficiently applying this operation leads to the latin square with diagonal entries filled with 0s. From Lemma 2, we know that a partially filled latin square containing all the 0 symbols is completable. Hence, we can always complete this partial latin square to obtain a valid network code.

Therefore the minimum energy network code for the node with two incoming edges is represented by latin squares with their main diagonal filled with 0 symbols. $\square$

Next, we show that there exists linear (at the channel symbol level) minimum energy network codes.

Theorem 2. *The linear network code*

$$f_u = au_1 + (M - a)u_2, \pmod{M} \tag{4}$$

is a minimum energy network code iff a is coprime with M.

Proof. It is shown in [21] that the mapping $au_1 + bu_2$ corresponds to a Latin square if and only if a and b are relatively prime with M. Consider the mappings of the form $au_1 + (M - a)u_2$. Then if a is relatively prime with M, $M - a$ is also relatively prime with M, since otherwise,

$$\exists k \in [M] \text{ s.t. } M - a = kM$$
$$a = (1 - k)M \text{(contradiction)}. \tag{5}$$

Additionally, whenever $u_1 = u_2$, $au_1 + (M - a)u_2 = 0$, filling the diagonal of the corresponding Latin square with 0 symbols, hence satisfying the minimum energy condition. $\square$

Corollary 1. *There exists $\varphi(M)$ # of linear minimum energy network codes, where $\varphi(.)$ is the Euler's totient function.*

Proof. Euler's totient function gives the maximum number of integers relatively prime with M that are less than or equal to M. If a is relatively prime, then $M - a$ is also relatively prime. Then $\varphi(M)/2$ of the relatively prime integers are smaller than $M/2$. Since we can obtain a different network code by switching a and $M - a$ (this is true since $M/2$ is not relatively prime with M), we have $\varphi(M)$ number of linear MENCs. $\square$

V. NETWORK ENERGY MINIMIZATION WITH MINIMUM ENERGY CHANNEL AND NETWORK CODES

In general, channel coding techniques are used to provide reliability for point-to-point communications. Channel codes use n-bit codewords as channel symbols, which are mapped to the source messages. This mapping is one-to-one for unique decodability. Hamming distance between two codewords is the number of bits that they differ in. The minimum Hamming distance among all the codewords of a code is called the *code distance*. Code distance is the main metric that is used to adjust the reliability of a channel code. With minimum distance decoding, the received *n-tuple* is mapped to the closest codeword in terms of Hamming distance. A code

corrects t errors, if it has code distance of $2t + 1$. Codes with larger distance are more reliable, as they can correct more errors. The set containing all the codewords is called *the codebook* and represented by $\mathcal{C}$. Hamming weight or simply the weight of a codeword is the number of non-zero symbols in the codeword. Weight enumerator of a codebook is the polynomial $W_{\mathcal{C}}(z) = \sum_i l_i z^i$, where l_i is the number of codewords with weight i.

In [16], we developed a novel energy minimizing channel code with controllable reliability, which is called minimum energy coding (MEC), and showed its suitability for nanoscale communications due to significant energy efficiency it provides compared to the classical block codes. Even though we have assumed OOK in our previous work due to limited complexity of nanoscale communications, MEC can also be used with PSK-like constellations. We choose the codeword set and sourceword-codeword mapping such that the average energy is minimized. In Sec. V-B, we combine minimum energy channel and network coding to minimize the total energy dissipated in the overall network. The details of MEC and its use in this work is given in Sec. V-A.

First, we review the minimum energy channel code, as proposed in [16]. Later, we obtain the analytical results regarding average energy per channel symbol for a relay node employing MEC and MENC.

A. Minimum Energy Channel Coding

Minimum energy coding (MEC) maps the messages to codewords such that the average codeword energy is minimized under asymmetric modulation assumption. Even though we assumed OOK in our previous work due to the complexity requirements of nanoscale communications, the same idea applies to M-ary modulations where a single symbol requires less energy than the others. However, since our aim is not to develop minimum energy codes for such modulations, we use MEC as it is proposed in [16]. With such modulation, where channel symbol with smaller energy cost is mapped to the 0 symbol, minimizing the average code weight is equivalent to minimizing the average energy dissipated for communications. Hence, in this section, we investigate the minimum average weight by noting that a constant, i.e., energy per high symbol, multiple of the minimum average weight yields the minimum average energy per codeword. Let the message set μ be given by $\mu = \{x_0, x_1, ..., x_{M-1}\}$ with cardinality M, X be the source random variable and $X = x_i$ be mapped to the codeword c_i. The following theorem is the main result of [16]:

Theorem 3. *Let x_i has probability $p_i \in \{p_0, p_1, ..., p_{M-1}\}$ and p_{max} be $\max(p_i)$. For a desired code distance d, the minimum expected codeword weight, $E[w]$ is*

$$\min(E[w]) = \begin{cases} (1 - p_{max})d, & p_{max} \geq 0.5, \\ \frac{d}{2}, & p_{max} < 0.5, \text{ if } d \text{ even}, \\ \lceil \frac{d}{2} \rceil - p_{max}, & p_{max} < 0.5, \text{ if } d \text{ odd} \end{cases} \tag{6}$$

where corresponding codebook has weight enumerator

$$W_{\mathcal{C}}(z) = \begin{cases} z^0 + (M - 1)z^d, & p_{max} \geq 0.5 \\ z^{\lfloor \frac{d}{2} \rfloor} + (M - 1)z^{\lceil \frac{d}{2} \rceil}, & p_{max} < 0.5. \end{cases}$$

Therefore, depending on the probability of the most probable source outcome, we either use the codeword set with all-zero codeword, i.e., $\mathcal{C}_0$, with weight enumerator $W_{\mathcal{C}_0} = z^0 + (M-1)z^d$ or the set with codewords of equal weight of $d/2$ (for even d), i.e., $\mathcal{C}_1$, with weight enumerator $W_{\mathcal{C}_1} = z^{\lfloor \frac{d}{2} \rfloor} + (M-1)z^{\lceil \frac{d}{2} \rceil}$. For both cases, the codeword with less weight is mapped to the source with probability p_{max}.

When we employ MEC in networks, the probability calculations are more tedious. It is important to discuss if we can choose different sets of codewords at each node u in the network. The best codebook selection for node u, i.e., the codebook minimizing the average transmission energy of u, depends on the probability of the most probable event as given in (6), for each edge $e \in \Gamma_{out}^u$. This cannot always be supported in networks, since any node u', where $e \in \Gamma_{in}^{u'}$ needs to distinguish between the selected codebooks, which requires extra bandwidth. Since our aim is to minimize energy, we assume that each source selects the codeword set minimizing its average codeword energy, and codebook selection can be inferred by the destination. In practice, the choice of codeword set can be indicated with a single bit, since only two codeword sets, i.e., $\mathcal{C}_0$ and $\mathcal{C}_1$ are sufficient for all the source distributions, as the choice only depends on whether p_{max} is greater or less than 0.5. Therefore, we assume that the selected codeword set is perfectly announced to the receivers without significant change in the performance.

B. Codebook Selection at the Relay

Each channel symbol $i \in \mathcal{A}$ represents a *length-n* codeword c_i composed of binary digits at the relay. Hence, there is a one-to-one and onto mapping between the channel symbols and codewords, i.e., $\xi : \{c_0, c_1, ..., c_{M-1}\} \rightarrow \mathcal{A}$, where $\xi(c_i) = i$. From Section V-A, we know that the selection between the codebooks $\mathcal{C}_0$ and $\mathcal{C}_1$ depends on the probability of the most probable source event. Let source events be sorted in decreasing probability order such that $p_i \geq p_{i+1}$. If probability of the most probable event, i.e., p_0, is greater than 0.5, codebook $\mathcal{C}_0$ is chosen, which assigns the all-zero codeword for c_0, and weight-d codewords for all the others, where d is the desired Hamming distance of the code. If $p_0 < 0.5$, either all codewords are weight-$d/2$ if d is even, or c_0 has weight $w_0 = \lfloor d/2 \rfloor$ and all other codewords have weight $w_i = \lceil d/2 \rceil$ if d is odd, with the selection of codebook $\mathcal{C}_1$. In addition to source nodes, this coding technique is also employed at the network coding node to minimize energy associated with the outgoing edges. Since either $\mathcal{C}_0$ or $\mathcal{C}_1$ is selected, without loss of generality, we can say that $weight(c_0) \leq weight(c_i), \forall i \in \mathcal{A}$.

Similar to MEC, for minimum energy network coding, we determine the codebook at the relay by checking the probability of the 0 symbol. We should consider the joint probability distribution of the incoming symbols at the relay to determine the outgoing symbol probabilities. Assume that the symbols at the relay inputs are independent. Let p_i and q_i be the probability that the symbol i is transmitted at the incoming edges e_1 and e_2, respectively. Since MENC is represented by an all-zero diagonal latin square, the probability of 0 symbol

at the relay, i.e., p_0^R, is given by $p_0^R = \sum_{i=0}^{M-1} p_i q_i$. Then we have the following theorem yielding the minimum average codeword weight obtained with network coding at the relay:

Theorem 4. *Let p_i and q_i be the probability that the symbol i is received at the incoming edges e_1 and e_2 of the relay, respectively. Then the minimum average codeword weight at the outgoing edge of network coding relay is given by*

$$\min(E_R[w]) = \begin{cases} (1 - \sum_i p_i q_i)\, d, & \sum_i p_i q_i \geq 0.5 \\ d/2, & \sum_i p_i q_i < 0.5, \text{ if } d \text{ even} \\ \lceil d/2 \rceil - \sum_i p_i q_i, & \sum_i p_i q_i < 0.5, \text{ if } d \text{ odd}, \end{cases} \tag{7}$$

where the corresponding codebook has the weight enumerator

$$W_{\mathcal{C}}(z) = \begin{cases} z^0 + (M-1)z^d, & \sum_i p_i q_i \geq 0.5 \\ z^{\lfloor \frac{d}{2} \rfloor} + (M-1)z^{\lceil \frac{d}{2} \rceil}, & \sum_i p_i q_i < 0.5. \end{cases}$$

Proof. Let $\tilde{p}_0 = \sum_i p_i q_i$. Following the proof of Theorem 3 in [17] by replacing $\tilde{p}_0$ with p_0, (7) follows immediately. $\square$

Therefore, relay node should know $\sum_i p_i q_i$ in order to select the codebook yielding the minimum average energy. In dynamical environments where source probabilities change in time, obtaining this term might not always be possible. In order to relieve the relay, we show that it does not need the probabilities of the incoming symbols, if $p_0, q_0 < 0.5$.

Lemma 4. *Let p_i, $i \in \{0, 1, ..., M-1\}$ be a probability distribution such that $0.5 \geq p_0 \geq p_1 \geq ... \geq p_{M-1} \geq 0$. Then,*

$$\arg \max_{(p_0, p_1, ..., p_{M-1})} \left(\sum_i p_i^2 \right) = (0.5, 0.5, 0, ..., 0). \tag{8}$$

Proof. Proof follows easily from the fact that the optimum of a convex maximization problem is at the boundary. $\square$

Lemma 5. *Let u be the coding node in In-Two network, and p_i and q_i be the probability that symbol i is received at the incoming edges e_1 and e_2 respectively, where $p_i \geq p_{i+1}$ and $q_i \geq q_{i+1}$, $\forall i \in \{0, 1, ..., M-2\}$. Then we have,*

$$\text{If } (p_0 \leq 0.5) \wedge (q_0 \leq 0.5) \Rightarrow \sum_{i=0}^{M-1} p_i q_i \leq 0.5. \tag{9}$$

Proof. From Cauchy-Schwarz inequality, we have,

$$\sum_i p_i q_i \leq \sqrt{\sum_i p_i^2} \sqrt{\sum_i q_i^2}. \tag{10}$$

From Lemma 4, we know that $\sum_i p_i^2 \leq 0.5$ if $p_i \leq 0.5$. Therefore, $\sum_i p_i q_i \leq 0.5$ if $p_i, q_i < 0.5$ $\square$

Hence, we conclude that if the most probable message of the source node is less that 0.5, none of the coding nodes in the network needs to know the probability distributions of incoming symbols to select the energy minimizing codebook, since then the probability of 0 symbol is always less than 0.5.

From a similar argument, we see that for the other cases, such a conclusion is not possible. For example, if $p_1 < 0.5$ and $q_1 > 0.5$, from Cauchy-Schwarz inequality, $\sum_i (p_i q_i) < \sqrt{0.5}$.

r_0	r_1	r_2	p_0
0	1	2	
2	0	1	
1	2	0	

r_0	r_1	r_2	p_1
2	0	1	
1	2	0	
0	1	2	

r_0	r_1	r_2	p_2
1	2	0	
0	1	2	
2	0	1	

Fig. 1. LENC for In-Three network with M=3.

	s_0	s_1	s_2	q_0
p_0	0	1	2	
	2	0	1	
	1	2	0	

s_0	s_1	s_2	q_1
2	0	1	
1	2	0	
0	1	2	

s_0	s_1	s_2	q_2
1	2	0	
0	1	2	
2	0	1	

	s_0	s_1	s_2	q_0
p_1	2	0	1	
	1	2	0	
	0	1	2	

s_0	s_1	s_2	q_1
1	2	0	
0	1	2	
2	0	1	

s_0	s_1	s_2	q_2
0	1	2	
2	0	1	
1	2	0	

	s_0	s_1	s_2	q_0
p_2	1	2	0	
	0	1	2	
	2	0	1	

s_0	s_1	s_2	q_1
0	1	2	
2	0	1	
1	2	0	

s_0	s_1	s_2	q_2
2	0	1	
1	2	0	
0	1	2	

Fig. 2. LENC for In-Four network with M=3.

VI. Low Energy Network Coding for In-N Networks

In-N networks are defined as the networks composed of network coding nodes with N incoming edges. As we investigate in Section IV-A, network codes for In-Two networks are equivalent to latin squares. Similarly, when there are more than two incoming edges, the object corresponding to the network code is an N-dimensional latin square, i.e., a *latin hypercube*. Next, we introduce the latin hypercube concept.

A. Latin Hypercubes as Network Codes

Consider an In-N network coding node. Reversibility condition on the network code implies that, knowing any $N-1$ number of inputs and the output, we should be able to deduce the remaining unknown channel symbol at the input. In other words, given any set of incoming symbols $\mathcal{U}_R \subset In(R)$ satisfying $|\mathcal{U}_R| = N - 2$, the network code mapping the unknown two incoming edge symbols to the outgoing edge symbol should be representable by a latin square. The object satisfying this condition is called a *latin hypercube*.

Definition: An N dimensional latin hypercube of order M is an M^N cube such that each symbol $i \in \{0, 1, ..., M - 1\}$ appears only once through each line, where a line is the set of values of the latin hypercube when all but one dimensions are fixed.

Consider the case with $N = 3$. The structure corresponding to a reversible network code becomes a *latin cube* with three dimensions. Apart from rows and columns, the dimension providing depth to the cube is called *the file*. Redefining the latin structure for cube, we say a latin cube is a 3-dimensional $M \times M \times M$ matrix containing all the elements of the set $\{0, 1, ...M - 1\}$ exactly once in each row, each column and each file. It is easier to visualize the structure for three dimensions, however, difficulties arise in comprehending the view of hypercubes with higher dimensions.

Due to the high dimensional nature of latin hypercubes, the elegant and simple findings of minimum energy network coding for In-Two networks cannot be obtained. As we see in the following sections, even in the three dimensional case, the minimum energy network code depends on the input probabilities. Therefore, we present a network code which provides low energy dissipation, if not minimum, for the general In-N networks. We develop *low energy network code* (LENC) by investigating the three dimensional case in detail.

B. Low Energy Network Coding - LENC

Assume that the source set cardinality $M = 3$ and the network coding node has three incoming edges, i.e., $N = 3$.

The reversible network code should be represented by a three dimensional latin cube of order 3. Let the probability mass function of the incoming flows be p_i, q_i and r_i, $i \in \{0, 1, ..., M - 1\}$, where the incoming symbols are x, y and z, respectively. An example latin cube representing the network code is given in Fig. 1. Each latin square gives the mapping of the network code for a given x value, e.g. , considering node u, the first latin square maps y and z to f_u, when $x = 0$.

Assume $p_0 > q_0, r_0$. From Section IV-B, we know that the latin square with the main diagonal filled with 0 symbol provides the minimum energy. Hence, we should reserve this latin square for the most probable event, i.e., for the largest p_i. Since p_i's are ordered such that $p_i \geq p_{i+1}$, the minimum energy latin square is reserved for p_0 as shown in Fig. 1. Moreover, since $p_0 > q_0, r_0$, this selection provides the largest reduction in average energy. However, it is not straightforward to select the other latin square assignments, since the values of p_i, q_i and r_i for $i \neq 0$ determine this weight-minimizing mapping. For example, the network code obtained by swapping the second and third latin squares provides lower average energy than the given one, if $(q_1 - q_2)(r_0 - r_1) < (q_0 - q_1)(r_1 - r_2)$.

The above network code is obtained with cyclic shifts of the rows. Note that the code value increases by 1 along z, and by 2 along x and y dimensions. This selection of increments is intentional to assure the reversibility of the network code. Moreover, the provided network code is linear. We define the generalization to more inputs and arbitrary source cardinality:

Definition: For a coding node u with N incoming edges with symbols $\{u_1, u_2, ...u_N\}$, and symbol set cardinality of M, if $p(u_\tau = 0) \leq p(u_i = 0), \forall i \neq \tau$, *the low energy network code* is,

$$f_u = u_\tau + \sum_{i \neq \tau} (M - 1)u_i \pmod{M}. \tag{11}$$

Lemma 6. *The low energy network code is reversible.*

Proof. A reversible network code should be representable by a latin hypercube. Therefore, given any $N - 2$ inputs, the mapping should be a latin square. Let the unknown inputs be (u_i, u_j). If $i, j \neq \tau$, the mapping is given by $f_u = c + (M - 1)u_i + (M - 1)u_j$, where c is a known constant. This
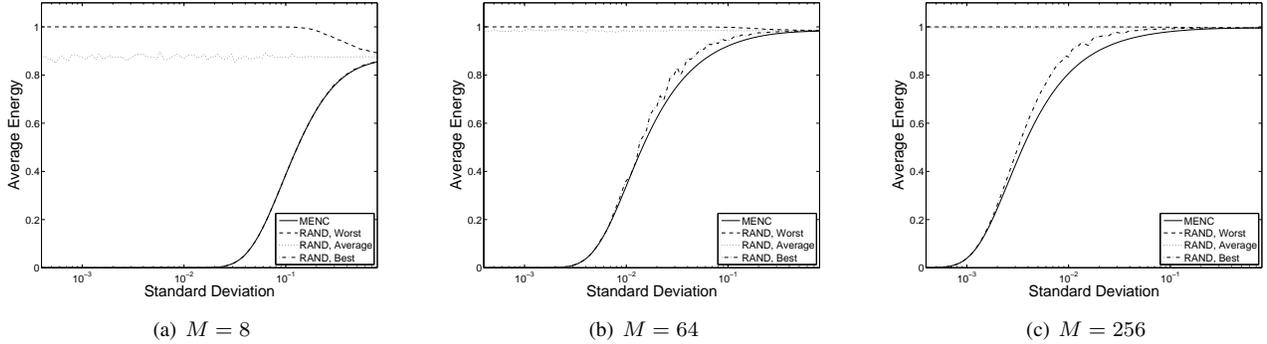
Fig. 3. Average energy to transmit 1.6 MB at the relay with MENC for varying standard deviation for different M values.

(a) $M = 8$ (b) $M = 64$ (c) $M = 256$

corresponds to a valid latin square, since $(M-1)$ is coprime with M (see proof of Theorem 2). If either i or j is τ, the mapping is the minimum energy latin square. $\qquad\square$

Lemma 6 assures that the LENC is representable by a valid latin hypercube. We aim to map minimum energy latin square, i.e., the latin square with all-zero diagonal, to the more probable incoming combinations by choosing the coefficient of $(M-1)$ for all but the least probable events. For example, if $p_0 > q_0, r_0$, the minimum energy latin square is assigned to the mapping of (u_2, u_3), when $u_1 = 0$, which is highly probable, leading to significant decrease in the average energy. The code provided in Fig. 1 is actually a LENC, for $r_0 < q_0, p_0$.

Another example LENC is shown in Fig. 2. Let u_1, u_2, u_3 and u_4 be the incoming edge symbols with pmf's of p_i, q_i, r_i and s_i respectively. Assume that $s_0 < p_0, q_0, r_0$. Then LENC is given by the mapping $f_u = u_4 + (M-1)(u_1 + u_2 + u_3) \pmod{M}$. To visualize the 4 dimensional latin hypercube corresponding to the LENC, latin squares corresponding to each (u_1, u_2) pair is demonstrated with varying u_3 and u_4.

VII. PERFORMANCE EVALUATION

In this section, we conduct numerical evaluations to illustrate the performance of MENC and LENC in In-Two and In-N networks for different source probability distributions.

A. Minimum Energy Network Coding

1) MENC vs. RAND for Asymmetric Energy Cost: Assume that the cost of transmitting a channel codeword c_i, i.e., ψ_i, is constant for all i's but $i = 0$, and $\psi_0 = 0$. We take an exponential probability distribution (pdf) with parameter σ in MATLAB, sample it M times and normalize to obtain a probability mass function (pmf). We investigate the average energy required at the relay node to transmit a 1.6 megabyte (MB) message. Assume that the energy per codeword depends logarithmically on M and is 1 microjoules (μJ) for $M = 256$.

Fig. 3 shows the variation of average energy required to transmit the message with respect to the standard deviation of source distribution for $M = 8$, $M = 64$, and $M = 256$. 1000 randomly filled latin squares representing random network codes are used. As observed in the figure, MENC reduces the average energy significantly compared to the average performance of random network codes, especially for small

σ, i.e., if the distribution is far from being uniform. MENC also achieves better performance than the best random network code out of 1000 trials. The fact that such performance is achieved with linear network codes is also important. Every network code perform the same for large standard deviation as expected, since the distribution becomes almost uniform.

2) MENC vs. XOR with MEC as the Channel Code: Second, we compare the proposed minimum energy network code for In-Two networks with the classical XOR network coding. It is important to note that, XOR network coding satisfies the minimum energy conditions for the binary field. However, with the employment of channel code to provide reliability, XOR, even though significantly reduces, does not minimize the average energy. We consider the scenario where the input probability distributions p_i, q_i satisfy the condition $p_0, q_0 > 0.5$, since for $p_0, q_0 < 0.5$, as shown in Section V-B, minimum average weight does not vary with the source pmf and can be achieved with MEC only.

Fig. 4 shows the minimum average weight comparison for XOR network code, random network code and MENC. The scenario with Hamming channel code instead of MEC is also provided for comparison. For fairness, MEC with $M = 16$ and $d = 3$ is compared to (7,4) Hamming, which also has code distance of 3. As expected, employing MEC at the source nodes greatly reduces the energy per codeword at the relay, since codewords are selected to minimize the average energy.

Note that different from Sec. VII-A1 MENC performs better for large σ, i.e., for more uniform distributions. This is due to that energy per channel symbol decreases with decreasing σ for both XOR and MENC, since each satisfies the minimum energy condition. However, XOR minimizes energy in the binary field, and further energy reduction is obtained with MENC when codeword length is greater than 2. Using MENC, network capacity can be achieved with almost half the energy required with XOR for uniform distributions.

It is important to underline that the reduction in energy obtained via minimum energy network coding is multiplied by the number of nodes in the network to obtain the overall energy reduction. Hence considering a large network and the fact that many codewords are required for a packet to be transmitted, this reduction in average codeword weight translates into drastic energy savings. For example, consider 1.6 MB of data. If we take $M = 8$, 3 bits are transmitted per codeword. Hence
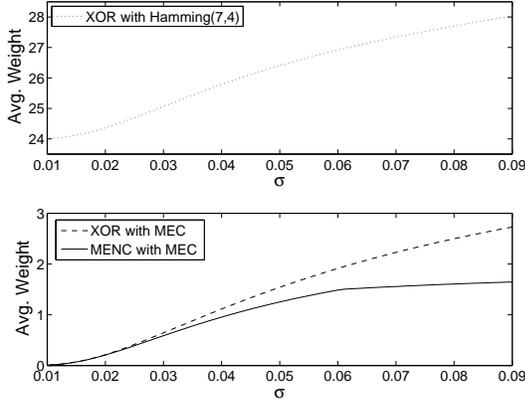
Fig. 4. Average weight per codeword at the relay for XOR with (7,4) Hamming, XOR with MEC, random network coding with MEC and MENC with MEC for In-Two networks.

Algorithm 1 Latin Hypercube Generation

$X_0 \leftarrow N - dimensional\ M^{th}\ order\ empty\ matrix$
$X \leftarrow X_0$
$i \leftarrow 0$
$j \leftarrow 0$
while $i < M^N$ **do**
  $\mathcal{M} \leftarrow \{0, 1, ..., M-1\}$
  flag $\leftarrow 1$
  while flag **do**
    for $j = 1$ to M **do**
      $r \leftarrow$ rand $(\mathcal{M})$
      $X(i) \leftarrow r$
      if isLatin(X, i, M, N) **then**
        flag $\leftarrow 0$
        $i \leftarrow i + 1$
      else
        $\mathcal{M} \leftarrow \mathcal{M} - \{r\}$
      end if
    end for
    if $j = M$ & flag **then**
      $i \leftarrow 1$
      flag $\leftarrow 0$
      $X \leftarrow X_0$
    end if
  end while
end while
return X

8.3^{12} number of codeword transmissions are required.

B. Low Energy Network Coding

Network codes for In-N networks are represented by latin hypercubes. To compare the performance of our low energy network code with random network codes, we should generate latin hypercubes in MATLAB. We first present our algorithms for latin hypercube generation. Later, we provide the performance comparison results for In-N networks.

1) Latin Hypercube Generation Algorithm: We use Algorithm 1 as a heuristic for latin hypercube generation. In the actual code, an additional control variable is employed, which stops the search if no progress is shown after a number of trials, which is omitted here for simplicity. *isLatin* is a simple
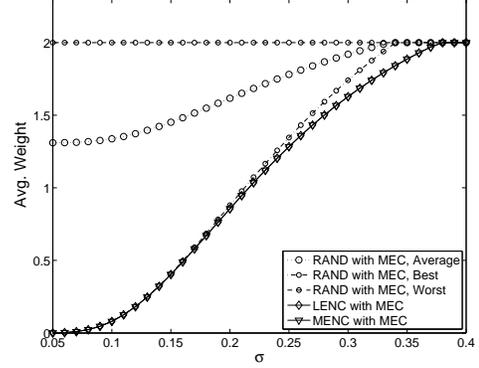

Fig. 8. Average energy per codeword for, random network coding, MENC and LENC for In-Three networks. $p_0, q_0, r_0 > 0.5$

algorithm (not shown) used to check if X, i.e., the high dimensional matrix representing the latin hypercube, still possesses the latin property after each step. Although not explicitly provided, the index M is used to obtain S_j. The proposed algorithm successfully generates randomly filled latin hypercubes with low dimension and small order. However, since the number of entries to fill increases drastically with increased dimension and order values, there is no guarantee that this algorithm will successfully generate large latin hypercubes. For example, despite significant wait time measured in days, we could not obtain any 3 dimensional latin hypercube of order 6 with this algorithm. Checking every hypercube is also not feasible since there are 6^{6^3} possibilities. The lack of any relevant results in the literature prevents us to provide performance results for large latin hypercubes. However, the essence of low energy network coding is successfully illustrated with the following results.

2) LENC vs. RAND for Asymmetric Energy Cost: Assuming normalized energy of 1 unit per codeword for all but one of the codewords, we compare the performance of LENC and random network code in terms of average energy per codeword. Fig. 5 illustrates the results for In-Three, In-Four and In-Five networks. For each dimension, we generate 100 random latin hypercubes of order 3 and obtain the minimum, average and maximum energy values corresponding to these random network codes. As observed in the figure, developed low-energy network code performs almost as good as the best random network code. Deterministically achieving such low energy values with LENC is of great importance. Moreover, this energy dissipation is achieved with linear network coding due to the linear nature of LENCs.

When the distribution has small standard deviation, most of the probability mass is accumulated towards the origin, i.e., the upper left symbols, of the latin hypercube. Therefore, the difference between the maximum and minimum energy values of random network codes increases. If the channel symbol with low energy dissipation overlaps with the high probability region of the latin hypercube, energy is greatly reduced, whereas in the opposite case, energy reduction due to asymmetric energy cost of the codewords is insignificant,
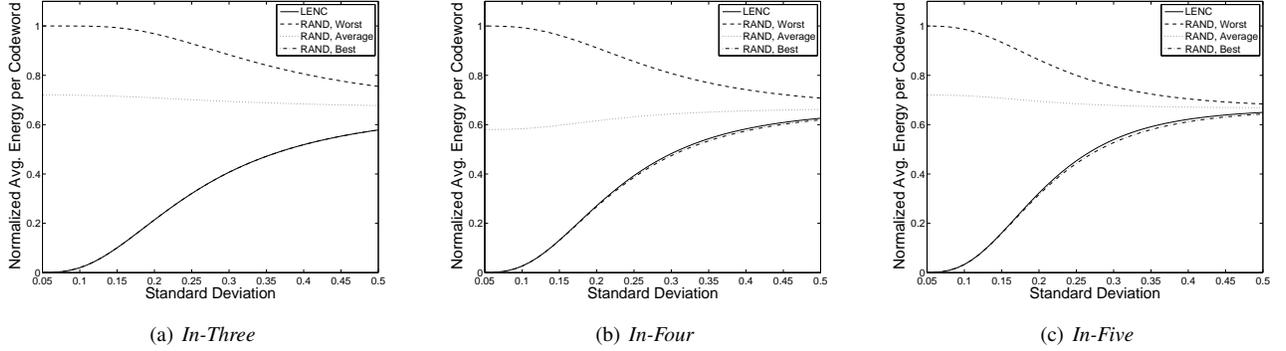
(a) *In-Three* (b) *In-Four* (c) *In-Five*

Fig. 5. Normalized average energy per codeword at the relay with LENC and random network code vs. standard deviation for different # of input edges.
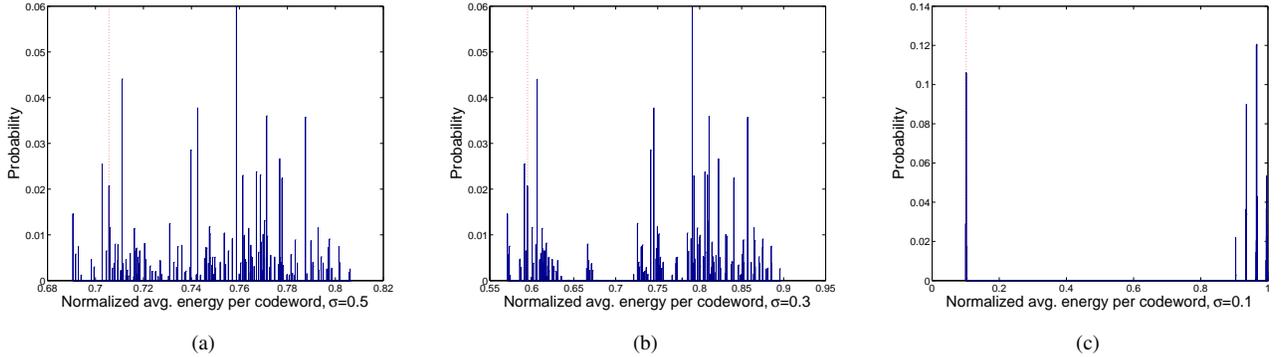

(a) (b) (c)

Fig. 6. Normalized # of random network codes (probability) and corresponding normalized average energy per codeword at the relay of *In-Three* network with random network code for $M = 4$ for standard deviation of $\sigma = 0.5, 0.3$ and 0.1. Red line shows the normalized average energy per codeword of LENC.
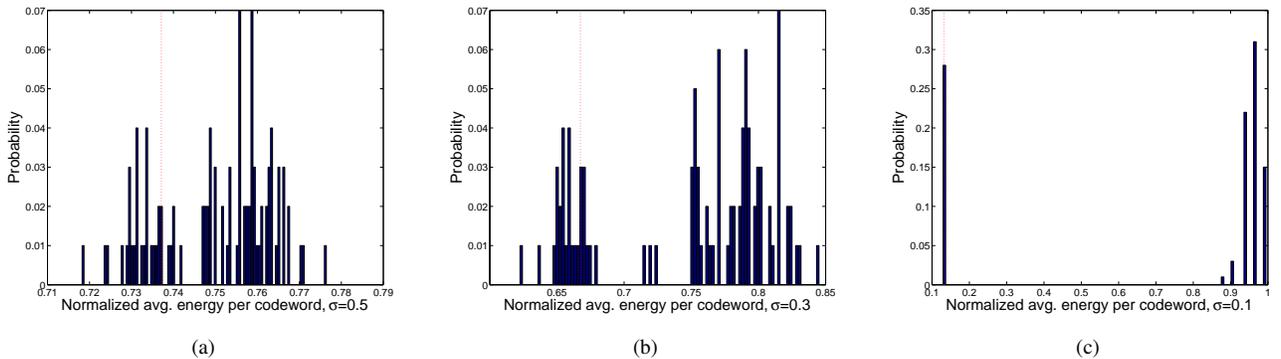

(a) (b) (c)

Fig. 7. Normalized # of random network codes (probability) and corresponding normalized average energy per codeword at the relay of *In-Four* network with random network code for $M = 4$ for standard deviation of $\sigma = 0.5, 0.3$ and 0.1. Red line shows the normalized average energy per codeword of LENC.

which causes the polarization of energy with random network coding, as observed in the figure. The dimension does not vary this behavior, but increased dimension requires smaller deviation to achieve significant energy efficiency.

3) RAND Histogram vs. LENC: To better illustrate the fact that LENC provides low average energy values compared to random network codes, and in an attempt to reveal the probability that energy associated with LENC is less than that of random network codes, we use average energy histograms of latin hypercubes in Fig. 6 and 7. The vertical red lines show where the corresponding LENCs stand in terms of energy.

Fig. 6 shows the number (normalized to probabilities over

the uniform measure on the observed latin cubes) of random network codes with corresponding average energy values out of 10000 random network codes for different standard deviation values for In-Three networks with $M = 4$. Smaller deviation polarizes the average energy between 0 and 1, as explained in Sec. VII-B2. As shown in the figure, probability that random network code provides smaller average energy than LENC is 0.121, 0.0688, and 0.0368 for $\sigma = 0.5$, $\sigma = 0.3$ and $\sigma = 0.1$, respectively, when network code is uniformly randomly chosen.

Similar results are obtained in Fig. 7, with 100 trials for 5th order In-Three and 4th order In-Four networks, respectively. It

is observed that, with increased dimension, smaller standard deviation values are needed to increase the probability that LENC performs better than the random network code, which is consistent with the findings in Sec. VII-B2.

4) LENC vs. RAND with MEC as the Channel Code: Since the probability calculations and determining MENC for In-N networks is tedious, we only consider In-Three networks with $M = 3$ to compare MENC with LENC. Fig. 8 shows the variation of the energy in 3th order In-Three networks. The underlying network code is LENC represented by the latin cube in Fig. 1. Performance of LENC is compared with MENC and random network coding. The random code is run 1000 times and the resulting minimum, average and maximum expected weights are given. As shown, LENC clearly outperforms the random network coding. Note that LENC corresponds to MENC in this case.

VIII. CONCLUSION

We developed network codes minimizing the average codeword energy for In-Two networks. The linear codes $f_u = au_1 + (M - a)u_2 \ \forall a \in [M]$ s.t. a is relatively prime with M satisfy the minimum energy conditions. Furthermore, we proposed a linear low energy network code for the general In-N networks and showed that it performs nearly as good as the best random network code in a deterministic manner. Simulations show that LENC provides much better performance in terms of average energy compared to random network codes.

REFERENCES

[1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. Yeung, "Network Information Flow," *in IEEE Trans. on Inf. Theory*, vol. 46, pp. 1204-1216, Jul. 2000.

[2] G. Wang, X. Dai, Y. Li, "On the Network Sharing of Mixed Network Coding and Routing Data Flows in Congestion Networks," *in IEEE Trans. on Vehicular Technology*, vol. 63, no. 5, pp. 2420–2428, Jun 2014.

[3] A. Antonopoulos, A. Lalos, M. Di Renzo, C. Verikoukis, "Cross-layer Theoretical Analysis of NC-aided Cooperative ARQ Protocols in Correlated Shadowed Environments," *to appear in IEEE Transactions on Vehicular Technology,* 2015.

[4] Jin Jin et al., "Multicast scheduling with cooperation and network coding in cognitive radio networks," *in Proc. IEEE INFOCOM*, 2010.

[5] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[6] R. Koetter, M.Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[7] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2745–59, Aug. 2005.

[8] R. Dougherty, C. Freiling, and K. Zeger, "Linear Network Codes and Systems of Polynomial Equations," *IEEE Transactions on Information Theory*, vol. 54, pp. 2303–16, May 2008.

[9] Y. Wu, P. A. Chou, and S. Kung, "Minimum-Energy Multicast in Mobile Ad Hoc Networks Using Network Coding," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1906–1918,Nov. 2005.

[10] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho, and E. Ahmed, "Minimum-cost multicast over coded packet networks," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2608–23, 2005.

[11] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," *in Proc. IEEE INFOCOM*, 2008.

[12] N. Gaddam, S. A. Gathala, D. Lastine and A. Somani, "Energy minimization through network coding for lifetime constrained wireless networks," *in Proc. IEEE ICCCN*, Aug. 2008

[13] L. Ding, P. Wu, H. Wang, Z. Pan and X. Yo, "Energy Minimization in Wireless Multihop Networks Using Two-Way Network Coding", *in Proc. IEEE VTC'2011*, 2011.

[14] A. Fiandrotti, V. Bioglio, M. Grangetto, R. Gaeta, E. Magli, "Band Codes for Energy-Efficient Network Coding With Application to P2P Mobile Streaming," *IEEE T. Multimedia*, vol. 16 , no. 2, pp. 521 – 532, 2014.

[15] C. Erin, H. H. Asada, "Energy optimal codes for wireless communications," *Proc. IEEE CDC'99*, Phoenix, USA, 1999.

[16] M. Kocaoglu, O. B. Akan, "Minimum Energy Coding for Wireless NanoSensor Networks," *in Proc. IEEE INFOCOM'12-Mini Conference*, Orlando, FL, Mar. 2012.

[17] M. Kocaoglu, O. B. Akan, "Minimum Energy Channel Codes for Nanoscale Wireless Communications," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1492 – 1500, Mar. 2013.

[18] M. Kocaoglu, D. Malak, "On the Node Density Limits and Rate-Delay-Energy Tradeoffs in Ad Hoc Nanonetworks with Minimum Energy Coding," *IEEE MoNaCom'12*, Canada, 2012.

[19] S. Zhang, S. C. Liew and P. P. Lam, "Hot topic: Physical-layer Network Coding", *ACM MobiCom '06*, pp. 358–365, Sept. 2006.

[20] B. Smetaniuk, "A new construction for Latin squares I. Proof of the Evans conjecture," *Ars Combin.*, vol. 11, pp. 155–172, 1981.

[21] P. Bartlett, Lecture notes, http://www.its.caltech.edu/~padraic/mathcamp_2012/latin_squares, 2012.

Murat Kocaoglu (S'06) received the B.S. degree in Electrical and Electronics Engineering with a minor in Physics from Middle East Technical University, Ankara, Turkey in 2010, and M.S. degree in Electrical and Electronics Engineering at the Next-generation and Wireless Communications Laboratory, from Koc University, Istanbul, Turkey in 2012. He is currently a PhD student at The University of Texas at Austin. His current research interests are in graphical models, learning theory and coding theory.

Ozgur B. Akan (M'00-SM'07) received the Ph.D. in electrical and computer engineering from Georgia Institute of Technology, Atlanta, in 2004. He is currently a Full Professor with the Department of Electrical and Electronics Engineering and the Director of the Next-generation and Wireless Communications Laboratory, Koc University, Istanbul, Turkey. His current research interests include wireless communications, nanoscale and molecular communications, and information theory.