This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

LDPC Codes for Network-coded Bidirectional Relaying with Higher Order Modulation

Karthik Ravindran, Andrew Thangaraj and Srikrishna Bhashyam
Department of Electrical Engineering
Indian Institute of Technology Madras, Chennai, India
Email: karthik.ravindran,andrew,skrishna@ee.iitm.ac.in

***Abstract*—We study the use of Low-Density Parity-Check (LDPC) codes for two-phase, network-coded bidirectional relaying with higher-order modulation. In the multiple-access phase, the sum of transmitted symbols scaled by the channel gains is the received relay constellation, which is network-mapped (clustered) to a transmit constellation for the ensuing broadcast phase. This operation at the relay is termed Clustered-Scaled-Sum (CSS) decoding. We propose a CSS coding scheme for bidirectional relaying using a single LDPC code over a ring with higher-order PAM or QAM alphabets. We design a message-passing decoder for CSS decoding with trade-offs possible between complexity and performance. We suggest a method for completing a Constrained Partially-filled Latin Square (CPLS) to a latin square, which is used in the construction of network maps at the relay for any channel fading state. The performance of the CSS coding scheme with LDPC codes over rings is shown to be very close to information-theoretic outer bounds.**

I. Introduction

The bidirectional (or two-way) relay problem has attracted significant attention in the past few years. The setting is shown in Fig. 1, where node A wishes to send a message to C, while node C has to send a message to A. The relay node B facilitates this exchange of information. We assume that there is no direct link between the two communication nodes A and C. Nodes are half-duplex, average power limited with receiver Additive White Gaussian Noise (AWGN) of variance σ_N^2. The gains of channels AB (also, BA) and CB (also, BC) are denoted h_1 and h_2, respectively. The pair (h_1, h_2) is said to be the channel fading state.

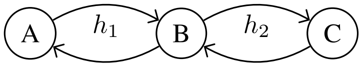


Fig. 1. Bidirectional relaying problem.

Bidirectional relaying was introduced in [1] [2] [3], and most of the evolution and prior work are summarized in the surveys [4] [5]. Though the exact capacity region is not known, schemes based on lattice coding have been shown to achieve rates within a small gap of the capacity region [6], [7]. In [6], a lattice coding scheme that achieves rates close to symmetric capacity for high SNR has been proposed for the $h_1 = h_2$ case. In [7], lattice coding schemes that achieve rates within $1/2$ bit for each direction and within $\log(3/2)$ bits of sum capacity are proposed for general (h_1, h_2). While the same lattice code is used at both nodes A and C in the $h_1 = h_2$ case in [6], different lattice codes are needed at nodes A and C for the general channel in [7]. In [8] [9], lattices over Eisenstein integers are studied. The relay attempts to decode a linear combination of transmitted symbols, where the coefficients are Eisenstein integers. The coefficients are chosen such that the achievable rate while decoding at the relay can be maximized. In [8], nested lattices are considered. Also, a practical implementation using Low-Density Parity-Check (LDPC) codes over a prime-sized field $\mathbb{F}_q$ is suggested. Standard message-passing algorithm for decoding over $\mathbb{F}_q$ is used. The modulation alphabet is chosen from a ring of Eisenstein integers. In [10], the authors study Physical-layer Network Coding (PNC) using nested lattice codes in an algebraic framework. Design examples based on lattice network codes in conjunction with convolutional and turbo codes are given.

Practical codes are usually designed based on finite alphabet constellations. Most of the initial code designs for bidirectional relaying used binary codes, binary modulation schemes, and XOR decoding at the relay [11]. However, higher order modulation schemes are required to achieve higher spectral efficiency. Recently, there has been an increased interest in the study of higher-order constellations [12]–[16] for bidirectional relaying. In [12], codes over higher-order fields have been employed. In particular, multiple network-coded linear combinations over fields are decoded at the relay. In [13], multi-level coding is used for compute-forward at the relay with network maps chosen using non-singular matrices over finite fields. The network coding map to be employed at the relay when higher-order constellations like M-PSK and M-QAM are used has been recently studied in [14]–[16]. The singular channel states that cause a collapse of the received alphabet are characterized in [15] and the channel conditions (h_1, h_2) for which the choice of network map is important are determined. While uncoded transmission is studied in [14]–[16], the network coding map is also required for coded transmission, and we use it appropriately in our work.

In [11], [17], [18], bidirectional relaying with coded modulation is studied. In [17], a non uniform M-PAM constellation is proposed along with the use of binary codes. But only the case of symmetric channels, where $h_1 = h_2$, is considered. Authors suggest the use of power control at the transmitting nodes in the case of asymmetric case, $h_1 \neq h_2$. But, this is possible only when the channel gains h_1, h_2 are known at the transmitting nodes.

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

In [18], the transmitting nodes use a convolutional code and binary phase shift keying for modulation. The relay decodes the XOR of messages using a Viterbi algorithm. In [11], RA codes are used at the transmitting nodes. The relay decodes the XOR of messages using a belief propagation algorithm. In both the cases, irrespective of the channel gains, a single linear function is decoded by the relay.

In this work, we propose and study a coding scheme based on Low Density Parity Check (LDPC) codes for network-coded bidirectional relaying with higher order modulation. Our design differs from existing designs in the following ways.

1) We consider the use of LDPC codes over *integer residue rings* (specifically ring of integers or Gaussian integers modulo 2^r) for the standard M-PAM and M^2-QAM constellations. This is unlike the constellations considered in [8], [9], which is based on lattices over Eisenstein integers, and in [17] which uses a non-uniform M-PAM.
2) We consider both linear and non-linear network maps for decoding at the relay. So far, the combination of coded modulation and non-linear network map in the context of bidirectional relaying is not reported in literature.
3) Since we also consider non-linear maps, the standard message-passing decoding algorithm over the ring $\mathbb{Z}_M$ cannot be used. We therefore propose the Clustered Scaled Sum (CSS) decoding algorithm.
4) In [15], the construction of a network map for a fading state is translated to a problem of completing a Constrained Partially-filled Latin Square (CPLS). While specific cases in which a CPLS could be completed are studied, no general methods exist for completing a CPLS with minimal number of distinct entries. In this work, we suggest randomized algorithms for completing a CPLS using weighted bipartite graph matching that attempts to minimize the number of distinct entries.
5) We design message-passing decoders that perform CSS decoding on the Tanner graph of the LDPC code. To ameliorate the complexity involved in decoding over higher-order rings, we propose modified decoders whose complexity and message size are reduced with a marginal decrease in performance.
6) Simulation results show that the proposed scheme can achieve: (1) rates close to the symmetric capacity for given (h_1, h_2), and (2) frame error rates close to the outage probability under a block fading model for (h_1, h_2).
7) Simulation results also show that regular LDPC codes over rings combined with an appropriate method for choosing network coding maps offer performance better than field-based schemes such as [12] and binary XOR-based schemes.

The rest of the paper is organized as follows. Clustering and network coding in a bidirectional relay are introduced in Section II. The CSS coding scheme with LDPC codes over rings and binary LDPC codes is described in Section III along with the message-passing CSS decoder. The completion of CPLS to construct a network coding map is discussed in Section IV followed by LLR compression in the CSS decoder in Section V. Simulation results are presented in Section VI followed by concluding remarks.

II. Clustering and Network Coding Map

Bidirectional relaying between nodes A and C happens in two phases - Multiple Access and Broadcast. In this section we consider the multiple access phase, in which nodes A and C are in transmit mode and the relay B is in receive mode. Let $\mathcal{A}$ denote the modulation alphabet. If node A transmits a symbol $s_1 \in \mathcal{A}$ and node C transmits a symbol $s_2 \in \mathcal{A}$, the received value at the relay is given by

$$y = h_1 s_1 + h_2 s_2 + z_B, \tag{1}$$

where z_B is additive Gaussian noise.

The symbol $h_1 s_1 + h_2 s_2$ received at the relay belongs to the received constellation

$$\mathcal{M}_B = \{s(u, v) = h_1 u + h_2 v : u, v \in \mathcal{A}\}.$$

The map $s(u, v) : \mathcal{A}^2 \to \mathcal{M}_B$ may be many-to-one, in general, and can cause a collapsing of points in $\mathcal{A}^2$ depending on the relative values of h_1 and h_2 and the modulation alphabet $\mathcal{A}$.

A. Network coding map

Following [14] [15], we define a network coding map or simply a network map $f : \mathcal{M}_B \to \mathcal{A}_{BC}$ ($\mathcal{A}_{BC}$ is the constellation used by the relay in the broadcast phase), which is said to be valid if it satisfies the following three conditions: (i) u can be uniquely computed given $f(s(u, v))$ and v, (ii) v can be uniquely computed given $f(s(u, v))$ and u, and (iii) superimposed symbols in $\mathcal{M}_B$ are mapped to the same symbol in $\mathcal{A}_{BC}$. While (i) and (ii) are important for final decoding in the broadcast phase at nodes A and C, (iii) is important for successful network-mapped decoding at the relay. The network map f is chosen based on the channel gains h_1, h_2. For every $u \in \mathcal{A}_{BC}$, the cluster $f^{-1}(u) = \{v \in \mathcal{M}_B : f(v) = u\}$ denotes the set of points in $\mathcal{M}_B$ that are all network-mapped to the same point in $\mathcal{A}_{BC}$.

One possible clustering and network map for 4-PAM are illustrated in Fig. 2 for some fading states with $\mathcal{A}_{BC}$ also set as 4-PAM. In Fig. 2, the values in $\mathcal{M}_B$ are shown below the axis, while the network-mapped values $f(\cdot)$ are shown on top. From Fig. 2, we see that the transmitted symbols are scaled, summed and clustered by the network map before decoding at the relay. Hence, the terminology Clustered-Scaled-Sum or CSS decoding.

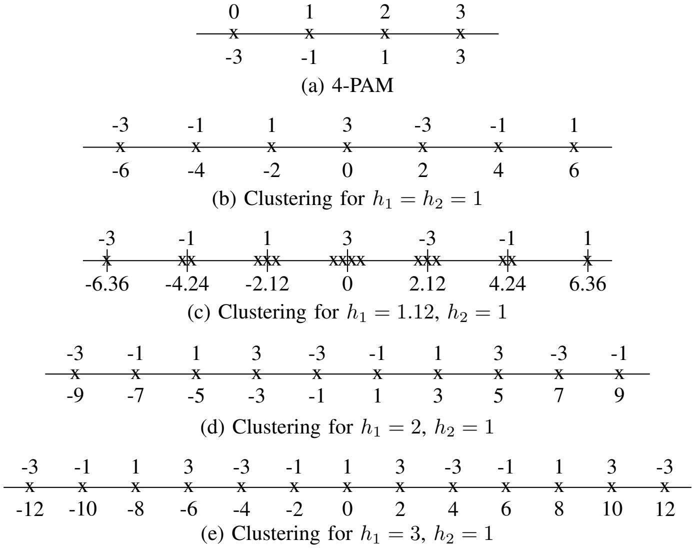
B. PAM/QAM constellations and integer residue rings

In this work, we consider the cases where the multiple access phase constellation $\mathcal{A}$ is either the standard M-PAM alphabet $\mathcal{P}(M) = \{-(M-1), \ldots, M-3, M-1\}$ or the M^2-QAM alphabet $\mathcal{Q}(M^2) = \{u + iv : \ u, v \in \mathcal{P}(M)\}$, where M is a power of 2.

For the broadcast phase, the size of the constellation needs to be $|\mathcal{A}_{BC}|$. If $\mathcal{A}$ is the M-PAM alphabet, we choose the broadcast constellation alphabet $\mathcal{A}_{BC}$ to be a lowest-energy, $|\mathcal{A}_{BC}|$-subset of the M_{BC}-PAM alphabet $\mathcal{P}(M_{BC})$, where $M_{BC} \geq |\mathcal{A}_{BC}|$. If $\mathcal{A}$ is M^2-QAM, we choose $\mathcal{A}_{BC}$ to be a lowest-energy, $|\mathcal{A}_{BC}|$-subset of the M_{BC}^2-QAM alphabet $\mathcal{Q}(M_{BC}^2)$, where $M_{BC}^2 \geq |\mathcal{A}_{BC}|$.

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications



(a) 4-PAM

(b) Clustering for $h_1 = h_2 = 1$

(c) Clustering for $h_1 = 1.12$, $h_2 = 1$

(d) Clustering for $h_1 = 2$, $h_2 = 1$

(e) Clustering for $h_1 = 3$, $h_2 = 1$

Fig. 2. Relay constellation and clustering for 4-PAM.

For the M-PAM or M^2-QAM constellations, the network map can be conveniently described using integer residue rings. Let $\mathbb{Z}_q = \{0, 1, \cdots q-1\}$ denote the ring of integers with addition and multiplication performed modulo q. The one-to-one map $s_P : \mathbb{Z}_M \to \mathcal{P}(M)$ with $s_P(a) = 2a - (M-1)$ is used to map symbol $a \in \mathbb{Z}_M$ to the PAM alphabet $\mathcal{P}(M)$. The map s_P is illustrated in Fig. 2(a) for 4-PAM, where the ring elements $\{0, 1, 2, 3\}$ are shown on top and the corresponding PAM alphabet values $\{-3, -1, 1, 3\}$ are shown at the bottom. The one-to-one map $s_Q : \mathbb{Z}_M[i] \to \mathcal{Q}(M^2)$ with $s_Q(a_x + ia_y) = s_P(a_x) + is_P(a_y) = 2(a_x + ia_y) - (1+i)(M-1)$ is used to map $a_x + ia_y \in \mathbb{Z}_M[i]$ to the QAM alphabet.

Similar to s_P, we define the one-to-one map $s'_P : \mathbb{Z}_{M_{BC}} \to \mathcal{P}(M_{BC})$ for the relay transmit constellation $\mathcal{A}_{BC}$. Since s_P and s'_P are one-to-one maps, any network map $f : \mathcal{M}_B \to \mathcal{A}_{BC}$ is equivalent to a map between rings $f_M : \mathbb{Z}_M^2 \to \mathbb{Z}_{M_{BC}}$ by setting $f_M(a, b) = s'^{-1}_P(f(s(s_P(a), s_P(b))))$. The map f_M is said to be a valid network map if it satisfies the following three conditions analogous to the three conditions for the map f: (i) Given a and $f_M(a, b)$, b can be uniquely determined, (ii) Given b and $f_M(a, b)$, a can be uniquely determined, and (iii) $f_M(a, b) = f_M(a', b')$ if $h_1 a + h_2 b = h_1 a' + h_2 b'$ for $a, b, a', b' \in \mathbb{Z}_M$.

Similar to the M-PAM case, for the M^2-QAM case, we define a one-to-one map $s'_Q : \mathbb{Z}_{M_{BC}}[i] \to \mathcal{Q}(M^2_{BC})$. So, any network map $f : \mathcal{M}_B \to \mathcal{A}_{BC}$ is equivalent to a map $f_{Mi} : \mathbb{Z}_M[i]^2 \to \mathbb{Z}_{M_{BC}}[i]$, and the criteria for validity for f_M can be readily extended to f_{Mi} as well.

For constructing a network map, we will equivalently design an integer residue ring function f_M or f_{Mi} satisfying the three properties required for validity. Methods for such constructions are given in [16] for some cases. We postpone the description of our proposed methods for constructing valid network maps to Section IV. In the next section, we assume that a valid network map is available and proceed to describe our coding scheme using LDPC codes.

III. LDPC Codes and CSS Decoding

Low Density Parity Check (LDPC) codes are standard today in many communication systems. LDPC codes over rings are well-known for their performance in higher order modulation over the point-to-point Gaussian channel [19]. In this work, we employ LDPC codes, both binary and over integer residue rings, for decoding the clustered-scaled-sum (CSS) of transmitted symbols in a bidirectional relay. Next, the coding setup with LDPC codes over the ring $\mathbb{Z}_M$ is described for M-PAM for simplicity. The extension to M^2-QAM is briefly mentioned later.

A. Coding setup for LDPC codes over rings

We target symmetric or equal rates of bidirectional relaying from both A and C. Interestingly, only a single LDPC code is used even though the channel gains h_1, h_2 are different. Using nested codes, the same coding setup can be extended to asymmetric rates. For simplicity, we describe only the symmetric-rate case.

Multiple access phase: An (n, k) LDPC code C over the ring $\mathbb{Z}_M$ is used at both the nodes A and C for encoding messages. Message vectors from nodes A and C are denoted $\mathbf{m}_1$ and $\mathbf{m}_2$, respectively, where $\mathbf{m}_i \in \mathbb{Z}_M^k$. The codewords at A and C are denoted $\mathbf{c}_1 = [c_{11} \ \cdots \ c_{1n}]$ and $\mathbf{c}_2 = [c_{21} \ \cdots \ c_{2n}]$, respectively, with $c_{ij} \in \mathbb{Z}_M$. After modulation at nodes A and C, we have the symbol vectors $\mathbf{s}_i = s_P(\mathbf{c}_i)$. The relay receives a vector $\mathbf{y}$ whose i-th symbol is given by

$$y_i = h_1 s_P(c_{1i}) + h_2 s_P(c_{2i}) + z_{Bi}, \tag{2}$$

where z_{Bi} is additive Gaussian noise. The relay uses a valid network map f_M and attempts to decode to the vector $\mathbf{c}_B = [c_{B1} \ \cdots \ c_{Bn}]$, where $c_{Bi} = f_M(c_{1i}, c_{2i}) \in \mathbb{Z}_{M_{BC}}$, using a message-passing decoder that is described in Section III-D. We refer to this decoding as CSS decoding. The relay's estimate after CSS decoding is denoted $\hat{\mathbf{c}}_B$. The multiple access phase is illustrated in Fig. 3.

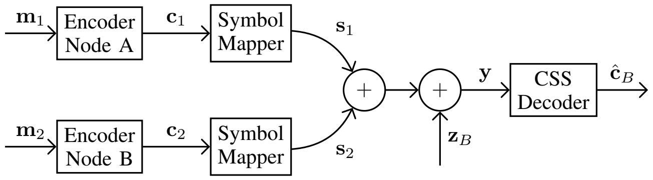


Fig. 3. Multiple access phase.

Broadcast phase: The estimated cluster-index vector $\hat{\mathbf{c}}_B \in \mathbb{Z}^n_{M_{BC}}$ is modulated to the vector $\mathbf{s}_3 = s'_P(\hat{\mathbf{c}}_B) \in \mathcal{A}^n_{BC}$, and is broadcast by the relay. Note that there is no further coding employed at the relay. As mentioned before, we choose $\mathcal{A}_{BC}$ to be a lowest-energy, $|\mathcal{A}_{BC}|$-subset of the M_{BC}-PAM alphabet $\mathcal{P}(M_{BC})$, where $\mathcal{M}_{BC}$ is chosen such that $\mathcal{M}_{BC} \geq |\mathcal{A}_{BC}|$. The received vectors at nodes A and C are denoted $\mathbf{y_A}$ and $\mathbf{y_C}$, respectively, with $\mathbf{z_A}$ and $\mathbf{z_C}$ being additive Gaussian noise vectors. The decoded vectors at nodes

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

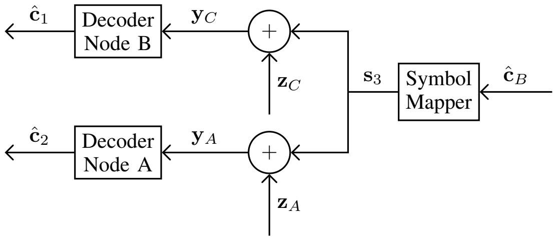


Fig. 4. Broadcast phase.

A and C are denoted $\hat{\mathbf{c}}_2 \in \mathbb{Z}_M^n$ and $\hat{\mathbf{c}}_1 \in \mathbb{Z}_M^n$, respectively. The broadcast phase is illustrated in Fig. 4.

The received values at A and C are

$$\begin{aligned} y_{Ai} &= h_1 s'_P(\hat{c}_{Bi}) + z_{Ai}, & (3)\\ y_{Ci} &= h_2 s'_P(\hat{c}_{Bi}) + z_{Ci}, & (4)\end{aligned}$$

for $1 \leq i \leq n$.

Decoding at A and C: Nodes A and C use a message-passing decoder for the code C. Decoder at node A takes as input, the probabilities $\mathrm{P}(c_{2i}|y_{Ai}, c_{1i})$, while, at node C, the input probabilities are $\mathrm{P}(c_{1i}|y_{Ci}, c_{2i})$, $1 \leq i \leq n$. Note that this is standard decoding of LDPC codes over rings [19].

As remarked earlier, we are targeting an equal rate of $(k \log_2 M)/2n$ bits per symbol from A to C and from C to A. Secondly, only a single code C over the ring $\mathbb{Z}_M$ is used in both the multiple access and the broadcast phase of the communication protocol for either encoding or decoding. Next, we characterize the information-theoretic achievable rate region of a two-way relay in which the relay does CSS decoding. This is useful for comparisons with simulation results.

B. Upper Bound on Achievable Rate Region

Let U, V be random variables corresponding to transmitted symbols from nodes A and C, respectively, each chosen uniformly at random from a transmit constellation $\mathcal{A}$. The received symbol at the relay is given by $h_1U + h_2V$, where h_1 and h_2 are gains of the channels AB and CB, respectively. Let $Y_B = h_1U + h_2V + Z_B$ be the random variable corresponding to the received value at the relay with Z_B being independent Gaussian noise. The relay uses a valid network map $f : \mathcal{M}_B \rightarrow \mathcal{A}_{BC}$ and attempts to decode $W = f(h_1U + h_2V)$.

As shown in the appendix, the achievable region in the MAC-phase is bounded on the outside by the region $\mathcal{R}_{MAC}$ defined as follows:

$$\mathcal{R}_{MAC} = \{r_x, r_y : \ 0 \leq r_x, r_y \leq I(W; Y_B)\}. \quad (5)$$

In the broadcast phase, to upper bound the achievable region, we suppose that the relay transmits W. Under this assumption, as shown in the appendix, the achievable rate region of the broadcast phase is bounded on the outside by the region $\mathcal{R}_{BC}$ defined as follows:

$$\mathcal{R}_{BC} = \{r_x, r_y \geq 0 : r_x \leq I(Y_A; W|U), r_y \leq I(Y_C; W|V)\}, \quad (6)$$

where $Y_A = h_1W + Z_A$ and $Y_C = h_2W + Z_C$ denote the received values at A and C, respectively, with Z_A and Z_C being independent Gaussian noise.

Let $\alpha \in [0, 1]$ be the fraction of channel uses for MAC-phase, and $(1 - \alpha)$ the fraction for the broadcast-phase. The achievable rate region with CSS decoding is bounded on the outside by the region $\mathcal{R}_{CSS}$ defined as follows:

$$\begin{aligned}\mathcal{R}_{CSS} = \{R_{AC}, R_{CA} : R_{AC} &\leq \min\{\alpha R_1, (1 - \alpha)R_2\},\\ R_{CA} &\leq \min\{\alpha R_3, (1 - \alpha)R_4\},\\ (R_1, R_3) \in \mathcal{R}_{MAC}&, (R_2, R_4) \in \mathcal{R}_{BC}\}. \quad (7)\end{aligned}$$

Note that the evaluation of $\mathcal{R}_{CSS}$, can be done numerically assuming suitable constellations for $\mathcal{A}$, $\mathcal{A}_{BC}$ and a network map f.

C. Review of LDPC decoding over $\mathbb{Z}_M$

For completion and ease of future description, we first describe a standard message-passing decoder for an LDPC code over the ring $\mathbb{Z}_M$. Note that this decoder is the one employed at nodes A and C in the broadcast phase. We will be brief and refer the reader to [19] for details. We describe the decoder in the probability domain, but there are equivalent descriptions and implementations in the log domain.

We assume that encoding is done with the LDPC code C over the ring $\mathbb{Z}_M$ with parity-check matrix H. The Tanner graph of H is a bipartite graph with two sets of nodes - Variable nodes and Check nodes. The variable nodes represent the columns of H, and the check nodes the rows. An edge exists between a variable node V_i and check node C_j if the (i, j)-th entry of H, denoted h_{ij}, has a non-zero value, with this value being called the weight of that edge. Note that h_{ij} will be assumed to be a multiplicatively invertible element or unit in the ring $\mathbb{Z}_M$.

Let us consider transmission over an AWGN channel with M-PAM modulation. The i-th received value y_i is given as

$$y_i = h s_P(c_i) + z_i, \ 1 \leq i \leq n, \quad (8)$$

where $c_i \in \mathbb{Z}_M$, z_i is additive Gaussian noise, and h is the channel gain. The channel input to the variable node V_i is the length-M probability mass function (PMF) $p_i^{(0)} = [\ \Pr(c_i = a|y_i) : a \in \mathbb{Z}_M]$, which is computed from the received value y_i.

Decoding takes place iteratively with messages passed between the variable and check nodes. The message passed between the variable node V_i and the check node C_j (in either direction) is a length-M PMF $[\Pr(c_i = a) : a \in \mathbb{Z}_M]$. The following three steps are involved in each iteration of decoding:

1) At a variable node, the outgoing PMF along an edge is computed as the element-wise product of all other incoming PMFs with suitable normalization. The incoming PMF from the channel $p_i^{(0)}$ is included at V_i, and, for the first iteration, there are no other incoming PMFs.
2) Because of the weights associated with the edges, the message vectors are permuted before and after every check node operation. The permutation π_{ij} that maps $a \in \mathbb{Z}_M$ to $h_{ij}a$ (well-defined because h_{ij} is a unit) is

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

applied to the PMF from variable node V_i to check node C_j, while π_{ij}^{-1} is applied in the reverse direction.

3) After the permutation, each check node imposes the parity check constraint that all incoming variables need to add to zero in the ring. Under this constraint, the outgoing PMF along an edge from a check node is computed as the one-dimensional discrete circular convolution of all other incoming PMFs along the other edges.

After every iteration, the output estimate of the PMF of c_i is computed at V_i as the normalized, element-wise product of all incoming PMFs. The decision $\hat{c}_i$ is set to be the ring element with maximum output PMF value.

D. CSS decoder over $\mathbb{Z}_M$

Clustered-Scaled-Sum (CSS) decoding at the relay is different from the standard decoding of LDPC codes described above. Consider the multiple access phase, where we have

$$y_i = h_1 s_P(c_{1i}) + h_2 s_P(c_{2i}) + z_i, \qquad (9)$$

where $\mathbf{c}_j = [c_{j1} c_{j2} \cdots c_{jn}]$, $c_{ji} \in \mathbb{Z}_M$, $j = 1, 2$ are the codewords transmitted, z_i is additive Gaussian noise, h_1, h_2 are the channel gains, and y_i is the received value.

In CSS decoding, the scaled sum, $h_1 s_P(c_{1i}) + h_2 s_P(c_{2i})$, is decoded up to a cluster defined by the network map f_M. This is equivalent to estimating the transmitted pair c_{1i} and c_{2i} in a way that pairs within a cluster resulting in the same $c_{Bi} = f_M(c_{1i}, c_{2i})$ need not be distinguished. The CSS Decoder, as shown in Fig. 5, takes as input, the joint PMF $\{\Pr(c_{1i} = a, c_{2i} = b | y_i)\}_{i=1}^n$ computed from the received values $\{y_i\}_{i=1}^n$ and gives an estimate of $\{c_{Bi}\}_{i=1}^n$.

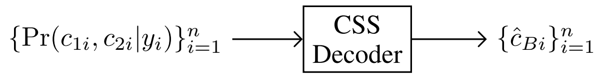

$\{\Pr(c_{1i}, c_{2i} | y_i)\}_{i=1}^n$
CSS Decoder
$\{\hat{c}_{Bi}\}_{i=1}^n$

Fig. 5. Input and output of a CSS Decoder.

The message passed between the variable node V_i and check node C_j is an $M \times M$ matrix, whose rows and columns are indexed by the M elements of $\mathbb{Z}_M$. The (a, b)-th entry is an estimate of the joint probability $\Pr(c_{1i} = a, c_{2i} = b)$. So, basically, we perform joint message-passing decoding of (c_{1i}, c_{2i}) in the Tanner graph.

Channel input: At variable node V_i, the channel input probability matrix is denoted S_0, and its (a, b)-th entry is computed as $\Pr(c_{1i} = a, c_{2i} = b | y_i)$ using (9). This is a fairly straightforward computation and we skip the details.

Variable node update: At every iteration, the variable node V_i receives joint PMFs S_1, S_2, …, S_d from the d (say) check nodes connected to it. Assuming that these messages are statistically independent, the outgoing message to the first check node is computed as the element-wise product

$$S_0 \odot S_2 \odot S_3 \odot \cdots \odot S_d, \qquad (10)$$

which is normalized to ensure that the sum of all entries is 1. Note that the (i, j)-th element of $A \odot B$ is $A_{ij} B_{ij}$. The other outgoing messages are computed using products similar to (10) with S_i omitted for the i-th check node.

Reordering on edges: The columns and rows of a message matrix S from variable node V_i to check node C_j are reordered to account for the multiplication by $h_{ij} \in \mathbb{Z}_M$. The permutation of the rows and columns is the permutation π_{ij} on $\mathbb{Z}_M$ induced by multiplication by the invertible element h_{ij}, and the element at (a, b)-th location of S is moved to the $(\pi_{ij}(a), \pi_{ij}(b))$-th location. In the reverse direction, the columns and rows of a message matrix S from check node C_j to variable node V_i are reordered with the permutation π_{ij}^{-1}.

Check node constraint: Since multiplication by h_{ij} is taken into account in the reordering, the constraints enforced by check node C_j are $\sum_{l \in N(j)} c_{1l} \mod M = 0$ and $\sum_{l \in N(j)} c_{2l} \mod M = 0$, where $N(j)$ denotes the set of variable nodes connected to C_j. It is readily seen that the message matrices are to be circularly convolved in two dimensions to enforce these constraints.

Check node update: At every iteration, the check node C_j receives messages S_1, S_2, …, S_d from the d (say) variables nodes connected to it. The outgoing message to the first variable node is computed as

$$S_2 \otimes_M S_3 \otimes_M \cdots \otimes_M S_d, \qquad (11)$$

where $\otimes_M$ denotes 2D circular convolution. The other outgoing messages are computed using products similar to (11) with S_i omitted for the i-th variable node.

Decision: For $a \in \mathbb{Z}_M$, the probability that c_{Bi} equals a is computed as

$$\Pr(c_{Bi} = a) = \sum_{(b_1, b_2) \in f_M^{-1}(a)} [S_0 \odot S_1 \odot S_2 \odot \cdots \odot S_d]_{b_1, b_2}$$

for a degree-d variable node v_i and $f_M^{-1}(a) = \{(b_1, b_2) \in \mathbb{Z}_M^2 : f_M(b_1, b_2) = a\}$ is the set of cluster points.

Stopping Criteria: If the estimated vector $[\hat{c}_{B1} \cdots \hat{c}_{Bn}]$ is same in two consecutive iterations or if the total number of iterations done equals a maximum number, decoding is stopped.

Extension to QAM: For the case where M^2-QAM constellation is used in the multiple access and broadcast phases, an (n, k) LDPC code over the ring $\mathbb{Z}_M[i]$ is used with the nonzero entries chosen as units in $\mathbb{Z}_M[i]$. The CSS decoder can be readily extended to the M^2-QAM case. The main difference is that the messages are $M^2 \times M^2$ joint PMFs with rows and columns indexed by elements of $\mathbb{Z}_M[i]$. The $(a_1 + ib_1, a_2 + ib_2)$-th entry of the message matrix represents the joint probability $\Pr(c_{1i} = a_1 + jb_1, c_{2i} = a_2 + jb_2)$.

In summary, we see that CSS decoding differs from standard LDPC message-passing decoding in the following ways: (1) The messages are joint PMFs of two transmit symbols - one from node A and the other from node C, (2) The check node constraints are on the individual transmit symbols, but are enforced on the joint PMF during update, (3) The stopping rule uses the network coding map as opposed to the correctness of the individual codewords.

E. Binary LDPC codes and CSS Decoding

In this section we consider the use of binary LDPC codes with higher order modulation with the relay doing CSS decod-

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

ing. We begin by briefly explaining the coding setup assuming that M-PAM constellation is used.

1) Coding setup for binary LDPC codes: Once again, we target symmetric rates and a single binary LDPC code is used for both phases of communication. The main modification in the binary case is the mapping of bits into Z_M-valued vectors and vice versa.

Multiple access phase: An (n, k) binary LDPC code C is used at both the nodes A and C for encoding messages. The binary codewords at A and C are denoted $\mathbf{b}_1 = [b_{11} \ \cdots \ b_{1n}]$ and $\mathbf{b}_2 = [b_{21} \ \cdots \ b_{2n}]$, respectively, with $b_{ij} \in \mathbb{Z}_2$. Let us define an arbitrary bits-to-symbol map $e_P : \mathbb{Z}_2^{\log_2 M} \rightarrow \mathbb{Z}_M$. With this map, each block of $\log_2 M$ bits within $\mathbf{b}_1$ and $\mathbf{b}_2$ is respectively mapped to get $\mathbb{Z}_M$-valued vectors $\mathbf{c}_1 = [c_{11}c_{12}\cdots c_{1n_s}]$ and $\mathbf{c}_2 = [c_{21}c_{22}\cdots c_{2n_s}]$, respectively, where $n_s = n/\log_2 M$ and $c_{ij} = e_P([b_{i(1+(j-1)\log_2 M)}\cdots b_{i(j\log_2 M)}])$, $i = 1, 2$ and $j = 1, 2, \cdots, n_s$. We assume that n is a multiple of $\log_2 M$. The relay receives

$$y_i = h_1 s_P(c_{1i}) + h_2 s_P(c_{2i}) + z_{Bi}, \tag{12}$$

for $1 \leq i \leq n_s$, where z_{Bi} is additive Gaussian noise. The relay uses a valid network map f_M and attempts to decode to the vector $\mathbf{c}_B = [c_{B1} \ \cdots \ c_{Bn_s}]$, where $c_{Bi} = f_M(c_{1i}, c_{2i})$, using a CSS decoder (adapted to binary codes) that is described in Section III-E2. The relay's estimate after CSS decoding is denoted $\hat{\mathbf{c}}_B$.

Broadcast phase: The estimated vector $\hat{\mathbf{c}}_B$ is broadcast to the nodes A and C in the broadcast phase after modulating to the broadcast constellation $\mathcal{A}_{BC}$.

Decoding at A and C: The received values at A and C are

$$\begin{aligned} y_{Ai} &= h_1 s_P(\hat{c}_{Bi}) + z_{Ai}, & (13) \\ y_{Ci} &= h_2 s_P(\hat{c}_{Bi}) + z_{Ci}, & (14) \end{aligned}$$

for $1 \leq i \leq n_s$, where z_{Ai} and z_{Ci} are additive Gaussian noises at A and C, respectively. Nodes A and C use a binary LDPC decoder for the code C. Decoder at node A takes as input, the probabilities $\Pr(b_{2i}|y_{Aj}, b_{1i})$, while at node C, the input probabilities are $\Pr(b_{1i}|y_{Cj}, b_{2i})$, where $1 \leq i \leq n$ and $j = \lceil \frac{i}{\log_2 M} \rceil$ provides the symbol index corresponding to i.

2) CSS Decoder with binary codes: The main modifications to the CSS decoder over higher order rings are the computations corresponding to the mapping and demapping operations. We will mostly emphasize the changes in our description of the decoder for the binary case.

In the binary case, iterative message-passing decoder takes place on the Tanner graph of the binary LDPC matrix. The channel input to the variable node V_i and message passed between the variable node V_i and check node C_j are 2×2 matrices, whose rows and columns are indexed by 0 and 1. These matrices are estimates of the joint PMF of bits b_{1i} and b_{2i} with the (β_1, β_2)-th entry being $\Pr(b_{1i} = \beta_1, b_{2i} = \beta_2)$.

Channel input: At variable node V_i, the channel input PMF is denoted S_0, and its (β_1, β_2)-th entry is computed as $\Pr(b_{1i} = \beta_1, b_{2i} = \beta_2|y_j)$, where $j = \lceil \frac{i}{\log_2 M} \rceil$ is the symbol index of the bit index i. This is computed by marginalizing the joint PMF $\mathrm{P}(c_{1j}, c_{2j}|y_j)$ using the bits-to-symbol map e_P. The joint PMF of (c_{1j}, c_{2j}) is, in turn, computed as

$$\begin{aligned} \Pr(c_{1j} = \alpha_1, &c_{2j} = \alpha_2|y_j) = \\ &\frac{p(y_j|c_{1j} = \alpha_1, c_{2j} = \alpha_2)P(\alpha_1, \alpha_2)}{p(y_j)}, \end{aligned} \tag{15}$$

where $P(\alpha_1, \alpha_2) = \Pr(c_{1j} = \alpha_1, c_{2j} = \alpha_2) = \frac{1}{M^2}$ for the first iteration, and $p(\cdot)$ denotes the PDF as per (12). For subsequent iterations, we use the estimated $\Pr(c_{1j} = \alpha_1, c_{2j} = \alpha_2)$ (17) from the previous iteration.

The variable and check node updates involve element-wise multiplication and the standard binary XOR rule for combining probabilities, and we omit the details.

Decision: The output joint PMF of (b_{1i}, b_{2i}) is given by the 2×2 matrix S^B computed as

$$S^B = S_0 \odot S_1 \odot S_2 \odot \ \cdots \ \odot S_d, \tag{16}$$

for a degree-d variable node V_i with incoming PMFs S_0 through S_d. Using the joint PMF of (b_{1i}, b_{2i}), the output joint PMF of $(c_{1j}, c_{2j}), 1 \leq j \leq n_s$, can be computed assuming that the bits within a block are independent. That is,

$$\Pr(c_{1j}, c_{2j}) = \prod_{k=(1+(j-1)\log_2 M)}^{j\log_2 M} \Pr(b_{1k}, b_{2k}), \tag{17}$$

where $c_{ij} = e_P([b_{i(1+(j-1)\log_2 M)}\cdots b_{i(j\log_2 M)}]), i = 1, 2.$ This assumption is clearly sub-optimal, and results in loss of performance as confirmed by simulation later.

Finally, for $a \in \mathcal{A}_{BC}$, the probability that c_{Bj} equals a is computed as

$$\mathrm{P}(c_{Bj} = a) = \sum_{(b_1,b_2)\in f_M^{-1}(a)} \mathrm{P}(c_{lj}, c_{kj}). \tag{18}$$

IV. Construction of Network Map

The choice of the network map $f : \mathcal{M}_B \rightarrow \mathcal{A}_{BC}$ depends on the fading state (h_1, h_2). Each possible valid network map f implies a specific clustering of the points of the received constellation at the relay $\mathcal{M}_B$. Following [20], we define *minimum cluster distance*, denoted $d_f(h_1, h_2)$, as the minimum of the cluster distances between all possible distinct pairs of clusters. That is,

$$d_f(h_1, h_2) = \min_{s,s'\in\mathcal{M}_B : f(s)\neq f(s')} |s - s'|. \tag{19}$$

Notice that the minimum cluster distance depends on f, h_1 and h_2. Clearly, for a given (h_1, h_2), it is best to pick a valid network map that maximizes $d_f(h_1, h_2)$, so that the clusters can be decoded at the relay with low probability of error. This maximization problem becomes complicated very quickly with increasing constellation size because of the difficulty in characterizing the validity of the network map analytically. To obtain a simple and intuitive solution, we use the following approach (that is later justified by simulations) based on singular fading states.

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

A. Singular fading states

The pair (h_1, h_2) is said to be a *singular fade state* [16] if the size of the relay constellation $\mathcal{M}_B$ is smaller than the maximum possible, i.e. $|\mathcal{M}_B| < |\mathcal{A}|^2$. Note that, up to scaling, singular fading states are finite in number for a given constellation and can be found easily [16].

Up to scaling and sign, the singular fading states for M-PAM correspond to $h_1 = l$, $h_2 = m$, where $l, m \in \{1, 2, \ldots, M-1\}$. Therefore, for singular fading states, we have $\mathcal{M}_B = \{lu + mv : u, v \in \mathcal{P}(M)\}$. Letting $u = s_P(a)$ and $v = s_P(b)$ for $a, b \in \mathbb{Z}_M$ and simplifying, we can write

$$\mathcal{M}_B = \{2(la + mb) - (l+m)(M-1) : a, b \in \mathbb{Z}_M\}. \quad (20)$$

For M^2-QAM, the singular fade states will correspond to $h_1 = l = l_x + il_y$, $h_2 = m = m_x + im_y$, where $l_x, l_y, m_x, m_y \in \{1, 2, \ldots, M-1\}$. Suppose M^2-QAM symbols $u = s_Q(a)$ and $v = s_Q(b)$ are transmitted from A and C. The relay receives $h_1 u + h_2 v$, which results in the constellation

$$\mathcal{M}_B = \{2(la+mb) - (l+m)(1+i)(M-1) : a, b \in \mathbb{Z}_M[i]\}. \quad (21)$$

In this work, we construct and store (at nodes A, B and C) valid network maps for each singular fading state. For an arbitrary fading state (h_1, h_2), the network map that maximizes the minimum cluster distance among the ones constructed for singular fading states is chosen. This search is finite and can be done offline and the results stored in look-up tables, if necessary. The chosen network map is assumed to be conveyed to nodes A and C, subsequently, before the broadcast phase.

The use of singular fading states has a direct effect on cluster distance. This can be seen from the example $(1.12, 1)$ shown in Fig. 2(c). Notice how the clustering for the nearby singular fading state $(1, 1)$ results in good minimum cluster distance for the state $(1.12, 1)$. If the fading state were far from a singular fading state, then the minimum cluster distance is likely to be good for many network maps. Therefore, clustering as per the nearest singular fading state (based on Euclidean distance) is an important heuristic for maximizing the minimum cluster distance, and we use this heuristic in the construction of network maps.

From now on, we focus on constructing valid network maps for singular fading states.

B. Linear network maps

Specifying the network map in terms of the ring $\mathbb{Z}_M$ for M-PAM and the ring $\mathbb{Z}_M[i]$ for M^2-QAM results in simplifications in some cases. For M-PAM singular fade states $h_1 = l$, $h_2 = m$, if l and m are both odd, we readily see that

$$f_M(a, b) = la + mb \mod M \quad (22)$$

is a valid network map. The conditions for validity are clearly satisfied because odd integers have multiplicative inverses modulo a power of 2. In addition to being a valid network map, the f_M in (22) is linear over $\mathbb{Z}_M$, which simplifies decoders at the relay. However, if either l or m is even, we need to construct network maps f_M that are non-linear over $\mathbb{Z}_M$. Examples of both the linear ($h_1 = h_2$, $h_1 = 3h_2$) and the non-linear case ($h_1 = 2h_2$) are illustrated in Fig. 2(b,d,e) for 4-PAM.

We note that, by considering rings of higher size ($\mathbb{Z}_R, R > M$), we can construct linear network maps for all singular fading states [21]. We also need R to be prime to ensure that the network map is valid. We may still need to consider non-linear network maps if it is required that the number of clusters in the received constellation at the relay, $\mathcal{M}_B$, is to be kept minimum.

For M^2-QAM singular fade states $h_1 = l$, $h_2 = m$ with $l, m \in \mathbb{Z}_M[i]$, consider the $\mathbb{Z}_M[i]$-linear map

$$f_{Mi}(a, b) = la + mb \mod M \quad (23)$$

with multiplication and addition performed over $\mathbb{Z}_M[i]$. The above linear map clearly satisfies the validity conditions if $l = l_x + il_y$ and $m = m_x + im_y$ are invertible in $\mathbb{Z}_M[i]$. This is true, whenever $l_x^2 + l_y^2$ and $m_x^2 + m_y^2$ are invertible modulo M. Since M is a power of 2, we see that l and m are invertible in $\mathbb{Z}_M[i]$ and f_{Mi} in (23) is a valid network map whenever $l_x^2 + l_y^2$ and $m_x^2 + m_y^2$ are both odd. However, if either of these are even, a non-linear (in $\mathbb{Z}_M[i]$) network map f_{Mi} needs to be used at the relay.

So, we see that, for a subset of singular fading states, a simple ring-linear network map is valid. We now move on to the more general case using latin squares.

C. Network maps using latin squares

The use of latin squares in the construction of network map for bidirectional relaying was suggested in [20]. In general, construction of a valid network map for a singular fading state reduces to completing a Constrained Partially-filled Latin Square (CPLS) [20]. Next, we define latin squares and describe its relevance for network map construction. Following this, we suggest methods for constructing a CPLS and for its completion to a latin square.

1) Latin squares in bidirectional relaying: A latin square of order n is a set of $n \times n$ cells. Each cell is identified using its row and column index, and has an associated value or entry in it. An entry can occur not more than once in any row or column.

Consider a bidirectional relaying setup in which nodes A and C use the constellation $\mathcal{A}$ during the MAC phase. The received constellation $\mathcal{M}_B$ and the network map $f : \mathcal{M}_B \to \mathcal{A}_{BC}$ can be associated to an order-$|\mathcal{A}|$ latin square with the rows and columns indexed by the points of $\mathcal{A}$. The row and column index of each cell corresponds to a symbol pair transmitted from nodes A and C, respectively. The entries of the latin square cell indexed by (u, v) is the network-mapped value $f(s(u, v)) \in \mathcal{A}_{BC}$ (we can equivalently choose the entries from the set $\{0, 1, \cdots |\mathcal{A}_{BC}| - 1\}$, which has a one-to-one map with the elements of $\mathcal{A}_{BC}$). By the conditions of validity of the network map, the above is readily seen to be a latin square.

Given a singular fading state (l, m), we consider the problem of constructing a valid network map as a latin square. We will start with an empty order-$|\mathcal{A}|$ square and start filling it with integer values starting from 0 such that a valid network

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

map is obtained. After the construction, we readily see that the number of distinct entries is equal to $|\mathcal{A}_{BC}|$, the size of the broadcast transmit constellation. So, an important goal in the construction is that the size of the set of entries for the latin square should be kept to a minimum. Since the network map needs to be valid, symbol pairs from A and C (u_1, v_1) and (u_2, v_2) resulting in the same relay constellation value $s(u_1, v_1) = s(u_2, v_2)$ should be assigned the same network-mapped value. In terms of latin squares, the cells (u_1, v_1) and (u_2, v_2) should be constrained to have the same entry. So, the construction of a latin square corresponding to a valid network map has two steps:

1) Construction of a Constrained Partially-filled Latin Square (CPLS).
2) Completion of CPLS to a full latin square.

From the literature, it is readily seen that completion of partially-filled latin squares and other similar problems are quite difficult [22]. Specifically, the general problem of completing a partially-filled latin square is known to be NP-complete. So, we do not obtain provably optimal algorithms for the above constructions. In most cases, our approach is to use randomized algorithms and repeat them for a sufficient number of times, and we have found this approach to be effective.

2) Construction of CPLS: For a given singular fading state, we construct the ordered set of subsets $L = \{S_i\}_{i=1}^{q}$, where $S_i \subset \mathcal{A}^2$ are sets of points that need to be network-mapped to the same value, i.e. for $(u_1, v_1), (u_2, v_2) \in S_i$, we have $s(u_1, v_1) = s(u_2, v_2)$. Also, these sets are ordered such that, $|S_1| \geq |S_2| \geq \cdots |S_q| > 1$. Note that we consider sets S_i that have at least two or more points. Cells corresponding to points in each set S_i need to be filled (as ordered in L) with the same entry chosen from $\{0, 1, \cdots q-1\}$ with the number of distinct entries kept to a minimum.

We have employed a simple randomized strategy for this construction and found it to be effective. We proceed in the order $i = 1, 2, 3, \ldots$. For $i = 1$, fill the cells in S_1 with 0. For the cells corresponding to S_i, one of the previous entries *is chosen at random* and accepted if the latin square conditions are not violated. If violated, we proceed to the next available new entry from $\{0, 1, \ldots, q-1\}$. This procedure is repeated for a sufficiently large number of times and the CPLS with minimal entries is retained. For good results, proceeding in the order of decreasing $|S_i|$ is observed to be important.

3) Completion of CPLS to a latin square: We describe two methods for completing the CPLS to a Latin Square. The first method is a simple extension of the method used for constructing the CPLS. The second method is based on weighted matching on bipartite graphs. In cases where most of the cells of CPLS are filled, both the methods perform equally well in attempting to complete to a latin square with minimum number of distinct entries. In other cases, the second method gives better results.

Method 1: While constructing a CPLS, we considered $L = \{S_i\}_{i=1}^{q}$ such that $|S_i| > 1, i = 1, 2, \cdots q$. That is, only the sets $|S_i|$ with at least two or more points were considered. If we also include the sets that have $|S_i| = 1$, the construction would lead to a latin square which is completely filled. That is, we consider $L = \{S_i\}_{i=1}^{q'}$ such that $|S_i| \geq 1, i = 1, 2, \ldots, q'$. The cell entries are chosen from $\{0, 1, \cdots, q'-1\}$, and we repeat the randomized construction retaining the latin square with minimum number of entries.

Method 2: In the second method, after the construction of a CPLS, we attempt to complete the partially-filled latin square row-wise, starting from the first row. It is easy to see that filling cells in a row with constraints on the possible values for each cell is similar to a problem of matching in a bipartite graph, with the left vertices representing the cells and the right vertices representing the possible entries. Each left vertex representing a cell is connected to a right vertex, if it represents a possible entry for that cell.

To minimize the number of distinct entries, the edges are given weights. Suppose we have completed filling of the first $r-1$ rows and currently at the r-th row. The edges connected to already used entries till the $(r-1)$-th row are given a lower weight and rest of the edges are given a higher weight. By performing minimum-weight bipartite matching, we give a higher preference to reusing entries and obtain latin squares with minimum number of distinct symbols. If there are multiple minimum-weight matchings for a row, we *sample a matching at random* with uniform probability [23]. If there are no matchings, the attempt to complete fails.

The whole process is run as a randomized algorithm starting from row 1 with an initial number of distinct entries, which we take to be $|\mathcal{A}|$ typically. After several attempts, if the completion fails, the number of distinct entries is increased and the randomized algorithm is run again. It is clear that this randomized algorithm will eventually succeed. The steps of the second method are summarized below.

1) Initialize the set of entries to be $\{0, 1, 2, \ldots, l-1\}$ with $l = |\mathcal{A}|$. Set $r = 1$.
2) For row r, construct a bipartite graph $G = (V_1, V_2, E)$, where the left vertices $V_1 = \{1, 2, \ldots, m\}$ represent the vacant cells in row r and the right vertices $V_2 = \{0, 1, \ldots, l-1\}$ correspond to entries. The edge $(i, j) \in E$ only if $j \in P_{ri}$, where P_{ri}, the set of possible entries for the i-th vacant cell, is the set-difference of $\{0, 1, 2, \ldots, l-1\}$ and the set of entries above the i-th vacant cell.
3) Assign weight w_1 to the edges $(i, j) \in E$, if j is used in rows 1 through $r-1$. Else, assign weight w_2.
4) If a matching exists, sample a minimum-weight matching at random, say using the method suggested in [23]. Increment r and go to step (2).
5) If a matching does not exist, reset $r = 1$ and increment number of attempts. If maximum attempts have been reached, increment l. Go to step (2).

The number of attempts depends on the size of the constellation A which is used at nodes A and C. For a 4-PAM constellation, we have used $w_1 = 1$ and $w_2 = 2$ with 1000 attempts. For 16-QAM, we use $w_1 = 1$, $w_2 = 2$ with 50000 attempts for good results.

D. Network map construction for 16-QAM

Now, we provide results for network map construction using latin squares for 16-QAM constellation. We have 388 singular

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

TABLE II. LATIN SQUARE CONSTRUCTED FOR $\frac{h_2}{h_1} = \frac{2+i4}{2+i6} = 0.7 - i0.1$ FOR 16-QAM.

13	12	9	6	5	15	4	14	16	7	11	10	8	2	1	3
1	7	2	14	12	11	5	16	13	10	9	8	15	6	3	4
11	10	14	8	1	13	9	4	2	16	12	15	3	5	6	7
8	15	12	11	2	5	16	7	3	6	10	14	1	4	13	9
16	13	15	1	11	9	10	2	6	5	14	3	12	7	4	8
4	14	13	2	8	16	15	3	7	12	5	1	9	10	11	6
15	8	10	16	3	12	13	6	1	14	4	7	2	9	5	11
3	9	16	10	4	14	11	15	5	2	1	12	6	8	7	13
12	16	5	3	6	10	7	1	15	11	8	2	4	13	9	14
2	1	3	4	16	7	12	5	11	9	15	6	13	14	8	10
6	3	4	13	7	2	14	11	8	15	16	9	5	1	10	12
5	6	7	15	9	8	2	12	10	4	3	13	11	16	14	1
14	2	1	7	10	4	3	8	9	13	6	5	16	11	12	15
7	4	8	9	13	3	6	10	12	1	2	11	14	15	16	5
10	11	6	5	14	1	8	9	4	3	13	16	7	12	15	2
9	5	11	12	15	6	1	13	14	8	7	4	10	3	2	16

fading states for 16-QAM, of which 44 cases allow for a valid linear network map. Following the procedure discussed in Section IV we obtain a valid network map for each singular fading state. The results are summarized in Table I. We see that out of the 388 network maps constructed, 346 maps have a broadcast constellation size $|\mathcal{A}_{BC}|$ less than or equal to seventeen. With reference to the procedure outlined in Section IV, we used 50000 attempts for the construction with weights $w_1 = 1$ and $w_2 = 2$. Table II shows a latin square that is constructed for 16-QAM constellation for the singular fading state $\frac{h_2}{h_1} = \frac{1+2i}{1+3i} = 0.7 - i0.1$. The network map constructed with this latin square groups points of the received constellation at the relay to form 16 clusters.

TABLE I. STATISTICS OF NETWORK MAP FOR 16-QAM.

Number of clusters $\lvert\mathcal{A}_{BC}\rvert$	Number of network maps
16	163
17	183
18	24
19	18

Next, we compare the effectiveness of Methods 1 and 2 in completing a CPLS to a latin square with minimum number of distinct entries. We consider the CPLS that are constructed for some of the singular fading states of a 16-QAM constellation.

We group the CPLS into different sets such that all the CPLS in a set have the same number of entries. The CPLS in each set are completed to latin squares using both the methods. For each set, we compute the average of the number of distinct entries in the completed latin squares. This is done for both the methods. The results are listed in Table III. The first column lists the number of entries in CPLS. The second and third column lists the average number of distinct entries in the completed latin squares for Method 1 and Method 2, respectively. For instance, the first row corresponds to the set of CPLS with 16 entries. To complete these CPLS to a latin square, Method 1 on an average requires 17.875 distinct entries, while Method 2 on an average requires 16.125 distinct entries. From Table III we infer that, as the number of entries in the CPLS increases, the gap between the two methods in terms of the number of distinct entries required decreases. Even though Method 1 is computationally less intensive than Method 2, for the case when most of the cells of CPLS are filled, the number of distinct entries required on average for Method 1 is same as that for Method 2.

TABLE III. COMPARISON OF METHOD 1 AND METHOD 2

No. of entries in CPLS	Average no. of distinct entries required, method 1	Average no. of distinct entries required, method 2
16	17.875	16.125
48	17.8929	16.6429
112	18	17
136	18	17.375
192	19	18.5
232	17	17
240	16	16

V. LLR COMPRESSION IN THE CSS DECODER

In typical implementations, storage space of edge messages dominates the power and area of LDPC decoders. For the CSS decoder of Section III-D, the message contains M^2 probabilities, which is unmanageable in implementations. Even for the LDPC decoders over $\mathbb{Z}_M$ that are used in nodes A and C in the broadcast phase, the message contains M probabilities, which is a major source of complexity. Therefore, it is important to consider sub-optimal implementations of decoders that use smaller message lengths.

A. *Simplifications from linear network map*

For cases, where the singular fading state admits a linear network map, such as the cases described in (22) and (23), the CSS decoder reduces to an LDPC decoder over the ring because the target codeword $[c_{B1} \ \cdots \ c_{Bn}]$, being a linear combination of the codewords $[c_{11} \ \cdots \ c_{1n}]$ and $[c_{11} \ \cdots \ c_{1n}]$, belongs to the LDPC code C. Since the network map chosen for a non-singular fading state is that of a nearby singular fading state, several fading states admit a linear network map, and the message length reduction from M^2 to M is effective for several cases.

For fading states that do not admit a linear network map over the ring (such as $h_1 = 2h_2$ for 4-PAM), the number of message probabilities remains at M^2. Also, LDPC decoders over $\mathbb{Z}_M$ involve M message probabilities, and it is desirable to reduce this to 2, which is the number for messages of binary LDPC codes. Note that two binary message probabilities adding to 1 reduce to one log likelihood ratio (LLR).

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

B. *Message compression*

In the message-passing decoders over rings (both CSS decoding and decoding LDPC codes), the message probabilities from a variable node to a check node tend to be sparse. However, because of the convolutions at the check node, the message probabilities out of the check node are more evenly distributed. To accommodate this difference, we suggest the following generic compression techniques at the variable node and check node.

Variable node messages: For messages from the variable node, the top two message probabilities, p_0 and p_1 are retained, scaled to $p_0/(p_0+p_1)$ and $p_1/(p_0+p_1)$ so that they add to one, and the other probabilities are set to zero. Note that this results in a single LLR message along with a binary vector to indicate the positions of the top two probabilities. The binary vector will be of length at most $2\log_2 M$ for LDPC decoding and length at most $4\log_2 M$ for CSS decoding.

Check node messages: For messages from the check node, the top two message probabilities are retained in proportion, and all the other message probabilities are assigned a fixed value p_c with the constraint that the sum of all probabilities is equal to 1. This requires a message with two probabilities and a binary vector to indicate the positions of the top two.

This method could be generalized to consider the top J variable node and check node messages. Accordingly, we also need a binary vector to indicate its positions.

VI. Simulation Results

In simulations, we use 4-PAM and 16-QAM constellations for illustration with suitable normalizations for unit energy. For coding, we use a $(3,18)$ regular LDPC code of block length $n=3600$ symbols chosen from $\mathbb{Z}_4$ and $\mathbb{Z}_4+i\mathbb{Z}_4$, respectively, for 4-PAM and 16-QAM constellations.

The construction of parity check matrix H is done as suggested in [19]. First we construct a binary parity check matrix for a $(3,18)$ regular code. Then we replace the non-zero entries of H with multiplicatively invertible elements from the rings $\mathbb{Z}_4$ or $\mathbb{Z}_4+i\mathbb{Z}_4$ based on the modulation alphabet. Encoding at nodes A and C using the matrix H is done as suggested in [24]. Though LDPC codes with optimized degree distributions are likely to perform better than regular codes, we find, as described below, that regular codes provide good performance at SNRs close to capacity outer bounds.

We denote the SNR of link AB as $\mathrm{SNR}_1=|h_1|^2/\sigma_N^2$ and that of link BC as $\mathrm{SNR}_2=|h_2|^2/\sigma_N^2$. The rate from A to C is denoted R_{AC} and the rate from C to A is denoted R_{CA}. We note that these rates are same as R_{AC}, R_{CA} used in equation (7). In our coding schemes, we typically set $R_{AC}=R_{CA}$. Since the LDPC code is a rate-$5/6$ code, with 4-PAM constellation we have a target rate $R_{AC}=R_{CA}=(5/6)\times 2/2=5/6$. Similarly, for 16-QAM constellation we have a target rate of $10/6$.

For obtaining capacity outer bounds on the rates (R_{AC},R_{CA}), we follow [25] [26] and use the following equations:

$$R_{AC},R_{CA} \;\leq\; \frac{C(\mathrm{SNR}_1)C(\mathrm{SNR}_2)}{C(\mathrm{SNR}_1)+C(\mathrm{SNR}_2)}, \qquad (24)$$

$$R_{AC}+R_{CA} \;\leq\; C(\min(\mathrm{SNR}_1,\mathrm{SNR}_2)), \qquad (25)$$

where $C(\mathrm{SNR})$ denotes the maximum capacity of an AWGN channel over all possible transmit constellations used in either the MAC or broadcast phase. The capacity outer bound for decode-and-forward strategy is compued as in [25]. The capacity outer bound with CSS decoding at the relay is computed as discussed in Section III-B.

A. *Singular fading states*

To validate our approach, we first consider singular fading states. In Fig. 6, we show a plot of the final symbol error rate at node A versus SNR_1 with 4-PAM constellation at the input for the singular fading states $h_1=h_2$ with the network map as given in Fig. 2(b), and for $h_1=3h_2$ with the map in Fig. 2(e). Since the network map is linear for $h_1=h_2$, we use an LDPC decoder over the ring $\mathbb{Z}_4$ at the relay. For the $h_1=3h_2$ case, since the network map is non-linear, we employ the joint CSS decoder. At a symbol error rate (SER) of 10^{-4}, the CSS decoder is about 2-2.5 dB from the capacity bound, which is shown as a vertical line, and is computed as the minimum SNR_1 that achieves the rate pair $(R_{AC},R_{CA})=(5/6,5/6)$ as per the capacity bounds given by (24) and (25). For the message-compressed decoder, we chose top-2 messages and the value $p_c=0.08$ (see Section V-B). The LLR-compressed decoder suffers a loss of only about 0.5 dB. The error rates at node B show similar behavior to that of node A.

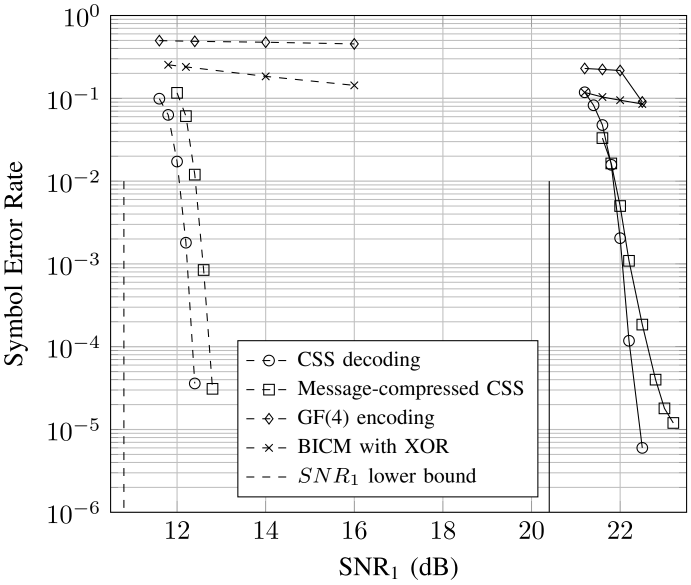


Fig. 6. Symbol Error Rate for 4-PAM, Dashed: $h_1=h_2$, Solid: $h_1=3h_2$.

We compare the symbol error rate of our scheme employing CSS decoding at the relay with the following two schemes: (1) Bit-Interleaved Coded Modulation (BICM) with XOR

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

decoding at the relay, (2) Encoding over the field GF(4) with the relay decoding the best scaled sum of transmitted symbols over this field [12]. These methods fail at the relay, as seen from the plot. This is because the XOR and GF(4) network maps have zero cluster distance for $h_1 = h_2$ and $h_1 = 3h_2$.

Note that all these simulations have been carried out with regular LDPC codes and at a moderate block length of 3600. Longer block lengths and codes with optimized degree distributions have the potential to provide near-capacity performance.

B. Fading channels and outage

We now consider the setting where the links AB and BC are considered as Rayleigh block fading channels. The channel gains follow a complex Gaussian distribution, $h_1, h_2 \sim \mathcal{CN}(0, \sigma_h^2)$. The average Signal-to-Noise ratio is computed as $\text{SNR}_i = \frac{\mathbb{E}(|h_i|^2)}{\sigma_N^2} = \frac{\sigma_h^2}{\sigma_N^2}, i = 1, 2$. Our proposed scheme employs CSS decoding at the relay with an LDPC code over the ring of length 3600. For each fading state, the network map is chosen to be the one among those for all singular fading states with the best cluster distance. For comparison, we consider the following schemes:

1) CSS decoding at the relay with the search for network maps restricted to only those that are linear over the corresponding ring. (This simplifies the decoder at the relay.)
2) Coding and network map over the field $\mathbb{F}_{16}$ [12].
3) Decode-and-Forward strategy.
4) CSS decoding at the relay with binary codes.
5) BICM with XOR decoding at the relay.

In Fig. 7, we plot the final frame error rate at node A versus the average SNR with 16-QAM constellation at the input. We now compare the performance of the above schemes at a FER of 10^{-2}. We see that the LDPC CSS scheme over rings (legend: CSS over ring) is at least 3 dB better than all other schemes. The clustering scheme that employs only linear network maps at the relay (legend: CSS ring-linear) is of interest to us since we can use a LDPC decoder at the relay instead of a joint decoder. This scheme is around 3 dB poorer when compared to the joint CSS decoding scheme. This is because, in many fading states, non-linear network maps offer higher minimum cluster distance for the received constellation at the relay. The avoidance of non-linear network maps causes a reduction in performance in the multiple access phase. We compare with the scheme [12], where encoding at nodes A and C are done over the field $\mathbb{F}_{16}$ (legend: GF(16) encoding). Here the relay attempts to decode all possible linear combinations of transmitted symbols over the field $\mathbb{F}_{16}$, and the decoding complexity at the relay is high. However, the performance is similar to that of the ring-linear network map case with a code over the ring, which decodes only one network map. The Decode-and-Foward (DF) scheme attempts to decode both the transmitted symbols from nodes A and C (legend: Decode-Forward). While the decoding complexity at the relay is similar to the proposed CSS decoding scheme, DF is about 5 dB away. The scheme with CSS decoding at the relay using binary codes (legend: CSS binary code) is about 5.5 dB away from the scheme with CSS decoding using code over the ring $\mathbb{Z}_4 + i\mathbb{Z}_4$. One of the reasons for the degraded performance is the independence assumption of bits, which is used by the CSS decoding algorithm on binary codes (used in (17)). BICM with XOR decoding at the relay (legend: BICM with XOR) has the least decoding complexity at the relay since a binary LDPC decoder is used, but is about 9 dB away from the proposed CSS decoding scheme. In many fading states, the network map based on bit-wise XOR may lead to a clustering of the received constellation at the relay that has zero or small minimum cluster distance. This leads to a poor decoding performance in the multiple access phase. So, the overall error rate is poor.

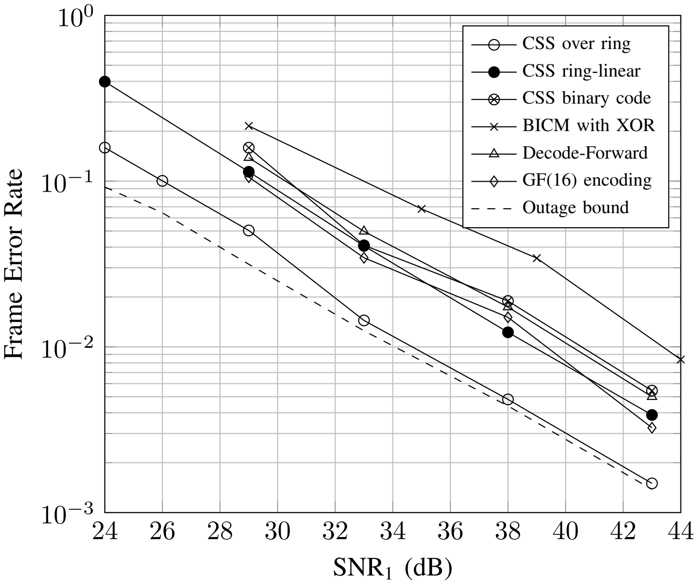


Fig. 7. Comparison of different schemes with 16-QAM, AB,BC being Rayleigh block fading channels.

Also, the outage bound is plotted for reference. This is calculated as the probability $\text{P}((R_{AC}, R_{CA}) \notin \mathcal{R}_B)$, where $\mathcal{R}_B$ is the region enclosed by the outer bounds (24), (25). At each SNR, we generate random values for the channel gains h_1 and h_2 following the Complex Gaussian distribution $\mathcal{CN}(0, \sigma_h^2)$, and the corresponding rate region enclosed by the outer bounds, $\mathcal{R}_B$, is computed. Since we use a rate-$5/6$ code with 16-QAM constellation, $R_{AC} = R_{CA} = \frac{5}{6} \times 4 \times \frac{1}{2} = \frac{10}{6}$. Outage is characterized as the event $(R_{AC}, R_{CA}) \notin \mathcal{R}_B$. The proposed CSS decoding scheme is around 1 dB away from the outage lower bound for a wide range of SNRs.

In Fig. 8, we compare the performance with the following decoders (at relay) of varying levels of complexity.

1) Joint CSS decoder without message compression
2) Joint CSS decoder with top-4 message compression
3) Joint CSS decoder with top-2 message compression
4) Linear network maps and LDPC decoder with top-4 message compression

The frame error rate versus the average SNR is plotted with 16-QAM constellation at the input when links AB and BC are

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

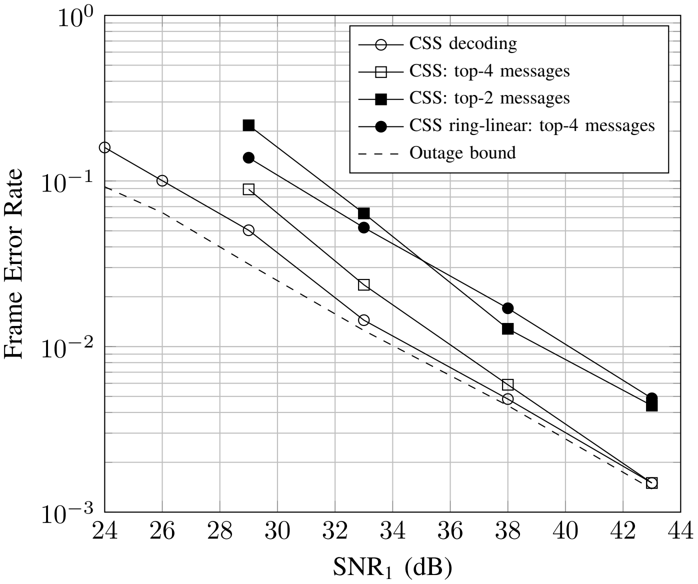


Fig. 8. Comparison of decoders with varying levels of complexity with 16-QAM, AB,BC being Rayleigh fading channels.

considered as Rayleigh block fading channels. The error rates of message-compressed CSS decoder with top-4 messages is close to that of the optimal CSS decoder at high SNRs. This is because, at high SNRs, the received values at the relay during the multiple access phase ((12)) are numerically close to values of the received constellation points (in $\mathcal{M}_B$) corresponding to the symbol pairs transmitted at nodes A and C. So, in the optimal CSS decoder, only few message probabilities of the message matrices $S_1, S_2, \cdots$ assume high values. Choosing the top 4 messages is usually sufficient at high SNRs. But, choosing only top-2 messages incurs a loss of about 5dB in comparison with the full-message CSS decoder. Using only ring-linear network maps and a LDPC decoder with top-4 messages at the relay, there is a loss of about 5.5dB in comparison with the full-message CSS decoder.

C. Rate Region

In Fig. 9, we plot the rate regions for QPSK and 16-QAM constellations for the singular fading states $h_2 = ih_1$ and $h_1 = ih_2$, respectively. The achieved rate pairs (R_{AC}, R_{CA}) at SER $= 10^{-4}$ are indicated at $SNR_1 = 6.3$ dB for QPSK and $SNR_1 = 12.8$ for 16-QAM. For QPSK, we use the ring $\mathbb{Z}_2[i]$. Also shown are outer bounds for Decode-Forward (DF) [25], the CSS scheme from (7) and an overall outer bound computed from ((24)- (25)). In the case of 16-QAM, the singular fading state allows for a ring-linear network map, and we used a message-compressed LDPC decoder with top-4 messages for the achieved rate. We see that for these specific fading states, the achieved rate pair is outside the DF outer bound. Also, notice that the achieved rate point is close to the CSS scheme outer-bound and the overall outer bound. This provides further justification to the good outage performance of the CSS scheme under fading conditions.

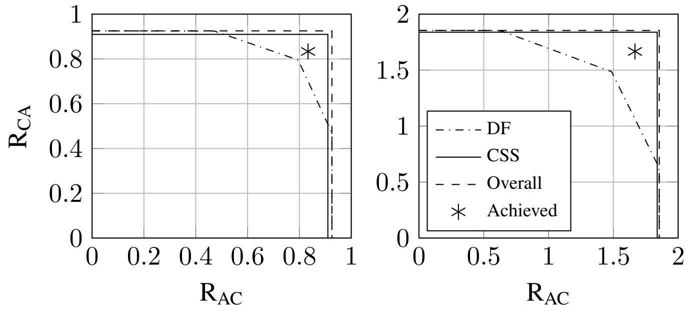


Fig. 9. Rate region for QPSK, $h_2 = ih_1$, $SNR_1 = 6.3$ dB and 16-QAM, $h_1 = ih_2$, $SNR_1 = 12.8$ dB.

VII. Conclusion

In this work, we proposed and studied the use of a Clustered-Scaled-Sum (CSS) coding scheme using LDPC codes over rings for bidirectional relaying with network coding and higher order modulation. We designed message-passing decoders suitable for CSS decoding at the relay. For network coding at the relay, we suggested a method for completing a Constrained Partially-filled Latin Square (CPLS) to a latin square with minimal number of entries. In the coded bidirectional relay setup, for an arbitrary channel state, we choose the network coding map as the one corresponding to a singular fading state that maximizes the minimum cluster distance.

We showed by simulation that the achieved rate points are close to the boundary of a capacity region outer bound. Further, the performance of the proposed CSS coding scheme with LDPC codes over rings is close to an outage probability lower bound under a block fading model and is significantly better than other schemes based on codes over fields and binary codes with BICM. The improvement in performance is mainly because we allow for the use of non-linear network maps and joint decoding over rings at the relay. To ameliorate complexity of joint decoding over rings, we propose LLR compression methods that reduce message size with marginal reduction in performance.

Appendix: Achievable Rate Region

We use standard notation and denote random vectors as $X^n = [X_1\ X_2 \cdots X_n]$ with X_i distributed *iid* according to a generic random variable X.

Let the vectors U^n and V^n transmitted from nodes A and C in the MAC phase be selected from codebooks of rate R_U and R_V, respectively. Let f be the network map, and let $W_i = f(h_1U_i + h_2V_i)$ be the network-mapped cluster index. The codebooks for U^n and V^n induce a codebook for W^n, whose rate is denoted R_W. A precise calculation of R_W is difficult, in general, and involves the structure of the codebooks and the network map. However, R_W clearly satisfies the inequalites $R_W \geq R_U$, $R_W \geq R_V$. Even though W^n is not directly transmitted, one can construct an equivalent channel where W is modulated into the relay constellation $h_1U + h_2V$ and transmitted over an AWGN channel as considered in [26]. So,

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2015.2433293, IEEE Transactions on Communications

the decoding of W is successful if

$$R_W \leq I(W; Y_B), \tag{26}$$

where Y_B^n is the random vector received at the relay B. Using the inequalities $R_W \geq R_U, R_V$, the achievable rate region in the MAC phase is seen to be bounded on the outside by the region $\mathcal{R}_{MAC}$ defined in (5).

To get upper bounds for the broadcast phase, we assume that the relay knows and transmits W^n. The received symbols are $Y_A = h_1 W + Z_A$ and $Y_C = h_2 W + Z_C$. We model the communication from B to A as an equivalent channel with state known at the receiver. In the equivalent channel, V is the transmitted symbol, and U is the channel state, which modifies the transmitted symbol to $W = h_1 U + h_2 V$. The channel state U is known at A. Using the well-known expression for capacity of channel with state known at the receiver [27], we get the outer bound $R_V \leq I(V; Y_A|U)$. Now,

$$\begin{aligned} I(V; Y_A|U) &\leq I(V, W; Y_A|U) \\ &\leq I(W; Y_A|U) + H(V|U, W) \\ &\leq I(W; Y_A|U), \end{aligned}$$

where the last expression follows because $H(V|U, W) = 0$ by the condition on the network map.

Using a similar model for the communicaiton from B to C as well, we get the upper bound in (6) for the broadcast phase.

References

[1] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '06, 2006, pp. 358–365.

[2] P. Popovski and H. Yomo, "The anti-packets can increase the achievable throughput of a wireless multi-hop network," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 9, 2006, pp. 3885–3890.

[3] B. Nazer and M. Gastpar, "Computing over multiple-access channels with connections to wireless network coding," in *Information Theory, 2006 IEEE International Symposium on*, 2006, pp. 1354–1358.

[4] P. Popovski and T. Koike-Akino, "Coded bidirectional relaying in wireless networks," in *New Directions in Wireless Communications Research*, V. Tarokh, Ed. Springer, Sep. 2009, pp. 291–316.

[5] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 438–460, March 2011.

[6] M. Wilson, K. Narayanan, H. Pfister, and A. Sprintson, "Joint physical layer coding and network coding for bidirectional relaying," *Information Theory, IEEE Transactions on*, vol. 56, no. 11, pp. 5641–5654, Nov. 2010.

[7] W. Nam, S.-Y. Chung, and Y. Lee, "Capacity of the Gaussian two-way relay channel to within bit," *Information Theory, IEEE Transactions on*, vol. 56, no. 11, pp. 5488–5494, nov. 2010.

[8] N. Tunali, K. Narayanan, J. Boutros, and Y.-C. Huang, "Lattices over Eisenstein integers for compute-and-forward," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, Oct 2012, pp. 33–40.

[9] Q. T. Sun and J. Yuan, "Lattice network codes based on Eisenstein integers," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, Oct 2012, pp. 225–231.

[10] C. Feng, D. Silva, and F. Kschischang, "An algebraic approach to physical-layer network coding," *Information Theory, IEEE Transactions on*, vol. 59, no. 11, pp. 7576–7596, Nov 2013.

[11] S. Zhang and S.-C. Liew, "Channel coding and decoding in a relay system operated with physical-layer network coding," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 5, pp. 788–796, June 2009.

[12] Z. Faraji-Dana and P. Mitran, "On non-binary constellations for channel-coded physical-layer network coding," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 1, pp. 312–319, 2013.

[13] B. Hern and K. Narayanan, "Multilevel coding schemes for compute-and-forward with flexible decoding," *Information Theory, IEEE Transactions on*, vol. 59, no. 11, pp. 7613–7631, Nov 2013.

[14] T. Koike-Akino, P. Popovski, and V. Tarokh, "Optimized constellations for two-way wireless relaying with physical network coding," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 5, pp. 773–787, 2009.

[15] V. Muralidharan, V. Namboodiri, and B. Rajan, "Wireless network-coded bidirectional relaying using latin squares for M-PSK modulation," *Information Theory, IEEE Transactions on*, vol. 59, no. 10, pp. 6683–6711, 2013.

[16] V. Namboodiri, K. Venugopal, and B. Rajan, "Physical layer network coding for two-way relaying with QAM," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 10, pp. 5074–5086, 2013.

[17] H. J. Yang, Y. Choi, and J. Chun, "Modified high-order PAMs for binary coded physical-layer network coding," *Communications Letters, IEEE*, vol. 14, no. 8, pp. 689–61, 2010.

[18] D. To and J. Choi, "Convolutional codes in two-way relay networks with physical-layer network coding," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 9, pp. 2724–2729, September 2010.

[19] D. Sridhara, "Bandwidth efficient coded modulation using low density parity check codes," PhD dissertation, University of Notre Dame, April 2003.

[20] V. Namboodiri, V. Muralidharan, and B. Rajan, "Wireless bidirectional relaying and latin squares," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, 2012, pp. 1404–1409.

[21] D. Fang, A. Burr, and J. Yuan, "Linear physical-layer network coding over hybrid finite ring for Rayleigh fading two-way relay channels," *Communications, IEEE Transactions on*, vol. 62, no. 9, pp. 3249–3261, Sept 2014.

[22] C. J. Colbourn, "The complexity of completing partial latin squares," *Discrete Applied Mathematics*, vol. 8, no. 1, pp. 25–30, 1984.

[23] J. Z. Broder, "How hard is it to marry at random? (on the approximation of the permanent)," in *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, 1986, pp. 50–58.

[24] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 638–656, 2001.

[25] S. J. Kim, N. Devroye, P. Mitran, and V. Tarokh, "Achievable rate regions and performance comparison of half duplex bi-directional relaying protocols," *Information Theory, IEEE Transactions on*, vol. 57, no. 10, pp. 6405–6418, October 2011.

[26] K. Ravindran, V. Boda, A. Thangaraj, S. Bhashyam, B. Joshi, and W. Li, "Optimized codes for bidirectional relaying," in *Communications (NCC), 2014 Twentieth National Conference on*, Feb 2014, pp. 1–5.

[27] A. E. Gamal and Y.-H. Kim, *Network Information Theory*. New York, NY, USA: Cambridge University Press, 2012.

0090-6778 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.