

# Network Coding with Computation Alignment

Naveen Goela<sup>†\*</sup>, Changho Suh<sup>‡\*</sup>, and Michael Gastpar<sup>†</sup>

**Abstract**—Determining the capacity of multi-receiver networks with arbitrary message demands is an open problem in the network coding literature. In this paper, we consider a multi-source, multi-receiver symmetric deterministic network model parameterized by channel coefficients (inspired by wireless network flow) in which the receivers compute a sum of the symbols generated at the sources. Scalar and vector linear coding strategies are analyzed. It is shown that *computation alignment* over finite field vector spaces is necessary to achieve the computation capacities in the network. To aid in the construction of coding strategies, network equivalence theorems are established for the decomposition of deterministic models into elementary sub-networks. The linear coding capacity for computation is characterized for all channel parameters considered in the model for a countably infinite class of networks. The constructive coding schemes introduced herein for a specific class of networks provide an optimistic viewpoint for the application of *structured codes* in network communication.

**Index Terms**—Vector linear network coding, computation capacity, computation alignment, structured codes.

## I. INTRODUCTION

Traditionally, network coding for wired networks with orthogonal links [1] has been successfully applied in multicast communication in part due to the optimality of linear network codes including both random and deterministic polynomial-time constructions [2]–[5]. However, for communication networks with general message demands, scalar and even vector linear codes were shown to be insufficient in characterizing capacity limits [6]. Without structure in the problem, there has been limited progress for inter-session network coding (see e.g., network coding with interference alignment for multiple-unicast demands [7]). This motivates the need to isolate the problem of shared network resources and multi-receiver demands by looking at simplified communication models for which new structured codes may be designed.

Recently coding for *computation* in networks has received considerable attention with applications to sensor networks. In [8] and [9], multiple-receivers in a sum-network compute the *sum of the symbols* generated at the sources. It is shown that there exists a linear solvably equivalent sum-network for any multiple-unicast network and vice-versa. The insufficiency of linear network coding for sum-networks is also established by correspondence to the result in communication networks, and an example of sum-network is given which is not solvable

over any field. In [10]–[12], Appuswamy et al. characterize the fundamental limits of computation in multi-source *single* receiver networks for linear and general target function classes. It is shown that the problem of computation in single-receiver networks is related (dual) to the multicast problem and hence tractable. In the present paper, we introduce structured network coding strategies based on *computation alignment* for multi-receiver network models where each receiver demands a sum of the source symbols. We simplify the problem by considering only a single-hop deterministic model inspired by its connection to wireless information flows.

Renewed interest in network coding has emerged recently due to the intimate connection between *wireless information flow* and wired networks. In [13], specific deterministic models are introduced such as the deterministic interference channel (IC) which closely approximates the Gaussian-IC in the limit of high signal-to-noise ratio. The deterministic network model is a generalization of networks with orthogonal links in the traditional network coding literature. Further connections between wireless communication and network coding were made in [14] where Nazer et al. introduced the compute-and-forward framework in which relay nodes in a network reliably compute functions of messages. Computation alignment over real vector spaces within the compute-and-forward framework was introduced in [15].

## II. NETWORK MODEL AND PRELIMINARIES

A sum-network is comprised of transmitters  $\mathcal{S} = \{1, 2, \dots, |\mathcal{S}|\}$  and receivers  $\mathcal{T} = \{1, 2, \dots, |\mathcal{T}|\}$ .

**Definition 1 (Source Information):** Each transmitter  $s \in \mathcal{S}$  observes a source message  $U_s \in \mathbb{F}_p^k$  in which each symbol  $U_s[j]$  for  $1 \leq j \leq k$  is drawn uniformly from a finite field  $\mathbb{F}_p$ .

**Definition 2 (Encoders):** Transmitter  $s$  uses encoder  $\mathcal{E}_s^{(n)}$  to map its message  $U_s$  to a length- $n$  codeword  $\{X_s[i]\}_{i=1}^n$  where  $X_s[i] \in \mathbb{F}_p^{L \times 1}$  is the channel input vector for the  $i^{\text{th}}$  channel use. The mapping over  $n$  channel uses is

$$\mathcal{E}_s^{(n)} : \mathbb{F}_p^k \rightarrow \mathbb{F}_p^{L \times n}.$$

**Definition 3 (Channel Model):** The channel model for the  $i^{\text{th}}$  channel use between transmitter  $s \in \mathcal{S}$  and receiver  $t \in \mathcal{T}$  is characterized by vector input  $X_s[i] \in \mathbb{F}_p^{L \times 1}$ , transfer matrix  $G_{s,t} \in \mathbb{F}_p^{L \times L}$ , and channel output  $Y_t[i] \in \mathbb{F}_p^{L \times 1}$ .

$$Y_t[i] = \sum_{s \in \mathcal{S}} G_{s,t} X_s[i]. \quad (1)$$

For this paper, we restrict attention to a class of symmetric linear deterministic models [13] for which  $G_{s,t}$  is the identity matrix for  $s = t$  and  $G_{s,t} = G_\gamma$  for  $s \neq t$ . The matrix  $G_\gamma \in \mathbb{F}_p^{L \times L}$  is a *downshift* matrix defined by parameters  $0 \leq \alpha \leq 1$  and shift  $\gamma \triangleq (\alpha - 1)L$ . The entries  $G_\gamma(i, j) = 1$  if  $i = j - \gamma$  and  $G_\gamma(i, j) = 0$  otherwise. The parameter  $\alpha$  is constrained

\* Equal contribution by both authors.

<sup>†</sup> N. Goela and M. Gastpar are with the School of Computer and Communication Sciences, Ecole Polytechnique Fédérale (EPFL), Lausanne, Switzerland (e-mail: {naveen.goela, michael.gastpar}@epfl.ch). N. Goela and M. Gastpar are also with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley, CA 94720-1770 USA (e-mail: {ngoela, gastpar}@eecs.berkeley.edu).

<sup>‡</sup> C. Suh was with the Research Laboratory of Electronics at the Massachusetts Institute of Technology (MIT), Cambridge, USA, and is currently with the Department of Electrical Engineering, KAIST, South Korea (e-mail: chsuh@kaist.ac.kr).

to be rational such that  $\alpha \triangleq \frac{m}{L}$  where  $m \in \mathbb{Z}^+$ . The network is characterized by parameters  $(m, L)$ .

*Example 1:* The channel model includes *broadcast and superposition* which allows the possibility for *in-network* computation.<sup>1</sup> In Figure 1, a sum-network is shown with  $|\mathcal{S}| = |\mathcal{T}| = 2$ , parameters  $(m, L) = (5, 6)$  and  $\alpha = \frac{5}{6}$ .

*Definition 4 (Decoders):* Each receiver  $t \in \mathcal{T}$  observes channel output vectors  $\{Y_t[i]\}_{i=1}^n$  over  $n$  channel uses. The goal for receiver  $t$  is to reconstruct  $\hat{V}_t$  using decoder  $\mathcal{G}_t$ :

$$\mathcal{G}_t^{(n)} : \mathbb{F}_p^{L \times n} \rightarrow \mathbb{F}_p^k,$$

$$\hat{V}_t = \mathcal{G}_t(Y_t[1], Y_t[2], \dots, Y_t[n]),$$

where  $\hat{V}_t \in \mathbb{F}_p^k$  is an estimate for the following linear function of the source symbols:  $V_t[j] = \sum_{s=1}^S \beta_s U_s[j]$  for  $1 \leq j \leq k$ . The function coefficients  $\beta_s \in \mathbb{F}_p$  and all operations are taken over  $\mathbb{F}_p$ . In this paper, we focus on *sum-networks* for which all coefficients  $\beta_s = 1$ .

*Definition 5 (Computation Rate):* A computation rate

$$R_{\text{COMP}} = \frac{k}{n} \log_2 p$$

is *achievable* in a sum-network if all receivers reliably compute the sum of source symbols, i.e. if for any  $\epsilon > 0$  and  $n$  large enough, there exist encoders  $\mathcal{E}_s^{(n)}$  and decoders  $\mathcal{G}_t^{(n)}$  such that  $\Pr\left(\left(\hat{V}_1, \dots, \hat{V}_T\right) \neq (V_1, \dots, V_T)\right) \leq \epsilon$ . The *computation capacity*  $C_{\text{COMP}}$  is the supremum of the achievable rates.

*Definition 6 (Linear Coding Capacity):* As in [6], let the *zero-error* network coding capacity  $C_{\text{COMP}}(f) = \sup\{\frac{k}{n} \log_2 p : \exists(k, n) \text{ network code that computes } f\}$ . A code computes  $f$  if  $\forall t \in \mathcal{T}$ ,  $\hat{V}_t = V_t$ . The linear coding capacity for computation  $C_{\text{COMP}}^{\text{lin}}$  is defined identically except that the network code consists only of linear mappings.

### III. MAIN RESULTS

*Theorem 1 (Linear Coding Capacity for Computation):*

Consider a sum-network with  $|\mathcal{S}| = |\mathcal{T}| = 2$ . Let  $p = 2$  with encoding and decoding operations over  $\mathbb{F}_2$ .

$$\frac{C_{\text{COMP}}^{\text{lin}}}{L} = \begin{cases} \alpha & \text{if } 0 \leq \alpha \leq \frac{2}{3}, \\ \frac{2}{3} & \text{if } \frac{2}{3} < \alpha < 1, \\ 1 & \text{if } \alpha = 1. \end{cases} \quad (2a)$$

$$\frac{2}{3} & \text{if } \frac{2}{3} < \alpha < 1, \quad (2b)$$

$$1 & \text{if } \alpha = 1. \quad (2c)$$

In the regime  $\frac{2}{3} < \alpha < 1$ , vector linear coding achieves higher computation rates than scalar linear coding.

*Corollary 1 (Limiting Capacity):* As established by Theorem 1, there exists a discontinuity at  $\alpha = 1$ .

$$\limsup_{\alpha \rightarrow 1^-} \frac{C_{\text{COMP}}^{\text{lin}}}{L} = \frac{2}{3}.$$

*Proof:* For a given  $(m, L)$  network, if  $L \bmod 3 = 0$  and  $\alpha \triangleq \frac{m}{L} = \frac{L-1}{L}$ , the construction given in Section V-A1 leads to a computation rate of  $\frac{R_{\text{COMP}}}{L} = \frac{2}{3}$ . This is the optimal rate assuming linear codes over  $\mathbb{F}_2$  as discussed in Section V-B. ■

<sup>1</sup>Several related types of linear deterministic channels have been studied as approximations to Gaussian channels and networks; see [13] or [16, Chap. 6.8] for further references.

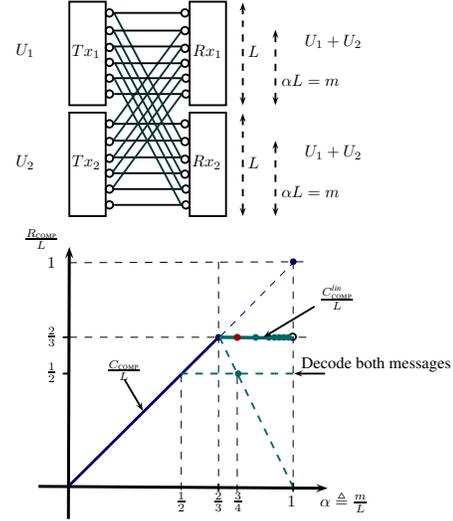


Fig. 1. (Top) A symmetric deterministic sum-network model  $(m, L) = (5, 6)$  with  $|\mathcal{S}| = 2$  transmitters and  $|\mathcal{T}| = 2$  receivers. For the  $i^{\text{th}}$  use of the memoryless channels, received signals  $Y_1[i]$  and  $Y_2[i]$  are linear shift combinations of the input channel signals  $X_1[i], X_2[i] \in \mathbb{F}_2^{L \times 1}$ . (Bottom) The achievable computation rate  $\frac{R_{\text{COMP}}}{L}$  is plotted for different channel parameters  $\alpha \triangleq \frac{m}{L}$  for a countably infinite class of networks.

*Remark 1:* For several types of multi-receiver networks, non-linear codes achieve higher rates than linear codes. However, in [17] we prove that Theorem 1 not only characterizes the linear coding capacity for computation, but indeed the full computation capacity over  $0 \leq \alpha \leq 1$ :  $\frac{C_{\text{COMP}}^{\text{lin}}}{L} = \frac{C_{\text{COMP}}}{L}$ .

### IV. PROOF OF THEOREM 1: PART I

For the regime  $0 \leq \alpha \leq \frac{2}{3}$ , we describe achievable coding strategies and the converse proof.

#### A. Coding Strategies

If both receivers decode both messages  $U_1$  and  $U_2$ , both receivers can also compute  $U_1 \oplus U_2$ . It can be shown that the multicast capacity for transmitting both messages to both receivers for an  $(m, L)$  network is given by  $R_1 \leq m$ ,  $R_2 \leq m$ , and  $R_1 + R_2 \leq L$ . Therefore, a lower bound on the computation rate is  $R_{\text{COMP}} \geq \min\{m, \frac{L}{2}\}$  which yields

$$\frac{R_{\text{COMP}}}{L} \geq \min\{\alpha, \frac{1}{2}\}. \quad (3)$$

Decoding both messages provides the optimal coding strategy for  $0 \leq \alpha \leq \frac{1}{2}$ . In order to achieve higher computation rates, the *channel structure* must be exploited for *in-network* computation. We introduce scalar linear network coding for the regime  $\frac{1}{2} \leq \alpha \leq \frac{2}{3}$ . The code construction is provided in Fig. 2. For the figure, we introduce the notation  $a_j \triangleq U_1[j]$  and  $b_j \triangleq U_2[j]$  for  $1 \leq j \leq k$  to represent the message bits of the first and second transmitter respectively. Another way of describing the linear code is via beamforming vectors. To send  $k$  message bits over  $n = 1$  channel use, both transmitters encode as follows:

$$X_1 = \sum_{j=1}^k a_j \Phi_j, \quad X_2 = \sum_{j=1}^k b_j \Gamma_j, \quad (4)$$

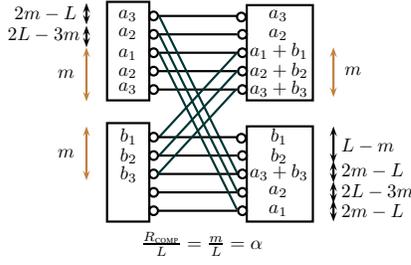


Fig. 2. Scalar linear code construction for  $(m, L)$  networks for  $\alpha \triangleq \frac{m}{L}$  in the regime  $\frac{1}{2} \leq \alpha \leq \frac{2}{3}$ . The specific example is for the  $(3, 5)$  model. The channel structure is exploited to compute the  $\oplus_2$ -function of message bits.

where  $\Phi_j \in \mathbb{F}_2^{L \times 1}$  and  $\Gamma_j \in \mathbb{F}_2^{L \times 1}$  are the beamforming or steering vectors for alignment in transmission. The code construction consists of designing all vectors  $\Phi_j$  and  $\Gamma_j$  at the transmitters (implicitly given in Fig. 2) so that the receivers may recover the computations  $a_j + b_j$ .

### B. Converse Proof

**Lemma 1 (Cut-set Upper Bound):** Due to the cut-set bound, the computation rate is limited by the total entropy of the output signals. The upper bound is  $\frac{R_{\text{COMP}}}{L} \leq 1$ .

**Lemma 2 (Converse in Regime  $0 \leq \alpha \leq \frac{2}{3}$ ):** A non-trivial upper bound in the regime  $0 \leq \alpha \leq \frac{2}{3}$  is given by

$$\frac{R_{\text{COMP}}}{L} \leq \alpha.$$

*Proof:* For the following steps, we write  $X_1^n$  and  $Y_1^n$  to mean  $\{X_1[i]\}_{i=1}^n$  and  $\{Y_1[i]\}_{i=1}^n$  respectively. By applying Fano's inequality,

$$\begin{aligned} n(R_{\text{COMP}}) &= H(U_1 \oplus U_2), \\ &= I(U_1 \oplus U_2; Y_1^n) + H(U_1 \oplus U_2 | Y_1^n), \\ &\leq I(U_1 \oplus U_2; Y_1^n) + n\epsilon_n, \\ &\leq I(U_1 \oplus U_2; Y_1^n, U_1) + n\epsilon_n, \\ &= I(U_1 \oplus U_2; Y_1^n | U_1, X_1^n) + n\epsilon_n, \\ &\leq H(Y_1^n | U_1, X_1^n) + n\epsilon_n, \\ &\leq \sum_i H(Y_{1i} | X_{1i}) + n\epsilon_n. \end{aligned}$$

The last inequality yields that  $n(R_{\text{COMP}}) \leq n\alpha L$  which proves the converse in the regime  $0 \leq \alpha \leq \frac{2}{3}$ . ■

## V. PROOF OF THEOREM 1: PART II

We prove that the linear coding capacity for computation is  $\frac{C_{\text{COMP}}^{\text{lin}}}{L} = \frac{2}{3}$  for all  $(m, L)$  networks such that  $\alpha \triangleq \frac{m}{L}$  is in the regime  $\frac{2}{3} \leq \alpha < 1$  by utilizing the following network equivalence and decomposition theorems.

### A. Decomposition into Parallel Deterministic Models

**Theorem 2 (Decomposition Theorem):** Any  $(m, L)$  model with  $\alpha \triangleq \frac{m}{L}$  and  $\frac{2}{3} \leq \alpha < 1$  may be decomposed into “Gap-1” parallel models of the form  $(r, r+1)$  where  $\frac{2}{3} \leq \frac{r}{r+1}$ .

*Proof:* See appendix for details. ■

**Corollary 2:** If a normalized computation rate of  $\frac{R_{\text{COMP}}^{\text{lin}}}{L} = \frac{2}{3}$  can be attained in any model of the form  $(r, r+1)$  for  $r \geq 2$

using linear coding, then a normalized computation rate of  $\frac{R_{\text{COMP}}^{\text{lin}}}{L} = \frac{2}{3}$  is achievable for any  $(m, L)$  model with  $\frac{2}{3} \leq \alpha < 1$ .

**Remark 2:** Ordinarily a decomposition into parallel models may not be helpful since coding *across* parallel models may be advantageous. For general parallel interference channels, there is no “separation” between parallel components.

**Example 2:** The following  $(m, L)$  network decompositions into orthogonal components hold as examples:<sup>2</sup>  $(7, 9) = (3, 4) \times (4, 5)$  and  $(17, 21) = (8, 10) \times (9, 11) = (4, 5)^3 \times (5, 6)$ .

**Theorem 3 (Computation Rate in “Gap-1” Models):** In any “Gap-1” model defined by  $(r, r+1)$  with  $r \geq 2$ , the computation rate achieved by linear codes is  $\frac{R_{\text{COMP}}^{\text{lin}}}{L} \geq \frac{2}{3}$ .

*Proof:* We prove Theorem 3 by classifying network models  $(r, r+1)$  with  $r \geq 2$  into three different cases:

- 1)  $(r+1) \bmod 3 = 0$ .
- 2)  $(r+1) \bmod 3 = 1$ .
- 3)  $(r+1) \bmod 3 = 2$ .

Each of these three cases is proved separately in the following sections.

1) *The Case of  $(r+1) \bmod 3 = 0$ :* For these networks, e.g.  $(2, 3)$ ,  $(5, 6)$ ,  $(8, 9)$ ,  $(11, 12)$ , only scalar network coding is necessary. The channel matrix  $G_\gamma$  for these “Gap-1” networks is a downshift matrix with  $\gamma = -1$ . Let  $k$  be the total number of message bits encoded by each receiver where  $k = \frac{2(r+1)}{3}$  to give a normalized computation rate of  $\frac{2}{3}$ . A scalar network code consists of beamforming alignment vectors  $\{\Phi_j\}_{j=1}^k$  for the first transmitter and  $\{\Gamma_j\}_{j=1}^k$  for the second transmitter. Let the notation  $\mathbf{e}_i \in \mathbb{F}_2^{r+1}$  denote the  $i$ -th coordinate vector. Consider the following paired alignments,

$$\begin{aligned} \Phi_1 &= \mathbf{e}_1, \Gamma_1 = G_\gamma \Phi_1 = \Phi_2, \Gamma_2 = \Phi_1, \\ \Phi_3 &= \mathbf{e}_4, \Gamma_3 = G_\gamma \Phi_3 = \Phi_4, \Gamma_4 = \Phi_3, \\ &\vdots \\ \Phi_{k-1} &= \mathbf{e}_{r-1}, \Gamma_{k-1} = G_\gamma \Phi_{k-1} = \Phi_k, \Gamma_k = \Phi_{k-1}. \end{aligned}$$

The above beamforming vectors allow *two* computations to be recovered at *both* receivers for each pair of consecutive alignment vectors  $\{\Phi_j, \Phi_{j+1}, \Gamma_j, \Gamma_{j+1}\}$  ( $j$  odd) which span *three* orthogonal subspace dimensions at each receiver. The interleaving pattern is repeated in multiples of three dimensions and tiles the whole space of  $r+1$  dimensions (where  $(r+1) \bmod 3 = 0$ ). The alignment applies to all networks of this type.

2) *The Case of  $(r+1) \bmod 3 = 1$ :* For these networks, e.g.  $(3, 4)$ ,  $(6, 7)$ ,  $(9, 10)$ ,  $(12, 13)$ , vector network coding is necessary and we show that  $n = 3$  channel uses is sufficient. Consider first the “indecomposable” model  $(3, 4)$  as an example. Figure 3 provides one linear code over  $n = 3$  channel uses which achieves a normalized computation rate of  $\frac{2}{3}$ . A total of 8 computations  $a_j + b_j$  are extracted at *both* receivers using only  $n(r+1) = 12$  transmitted symbols.

For general network models in this class, we construct a vector linear code over  $n = 3$  channel uses. It is observed (via network equivalences) that vector coding with  $n = 3$  for

<sup>2</sup>The symbol  $\times$  denotes the concatenation of orthogonal models as by analogy to the mathematical notation  $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$ .

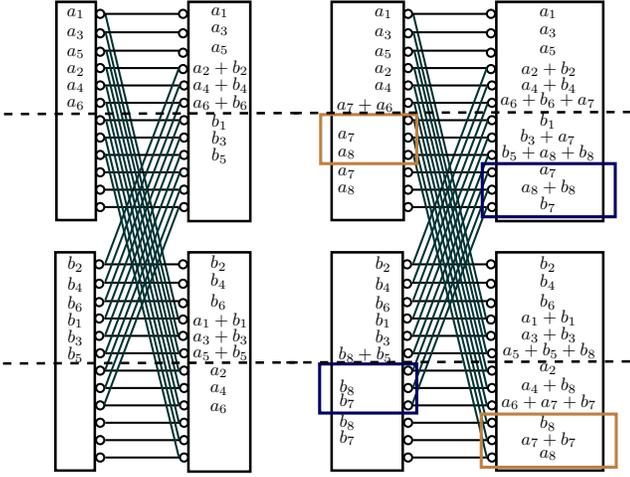


Fig. 3. Explicit scalar linear code for the  $(m, L) = (9, 12)$  model. The code corresponds to a *vector linear code* over  $n = 3$  channel uses of the  $(3, 4)$  model. The computation rate achieved is optimal  $\frac{R_{comp}}{L} = \frac{8}{12} = \frac{2}{3}$ , made possible by *computation alignment*. On the left, a total of 6 computation bits are possible using alternating computation alignment. On the right, 2 more computation bits for both receivers are possible due to careful alignment of beamforming vectors within the received dimensions.

an  $(r, r+1)$  model is identical to scalar linear coding over the model  $(3r, 3(r+1))$ . As an example, vector coding with  $n = 3$  for the model  $(6, 7)$  is equivalent to scalar linear coding for the model  $(18, 21)$ . We now show that linear code construction for the  $(18, 21)$  model follows in a straightforward manner from the achievable strategies already presented. On the left side of Figure 3, we observe that 6 computations are achieved over a total of 9 dimensions at both receivers. Subtracting these 9 dimensions completely from the  $(18, 21)$  model exactly “resets” the network model to a  $(9, 12)$  model for which we know that using the code construction given in Figure 3, as many as 8 computations are possible. Thus, a total of  $8 + 6$  computations are possible in the  $(18, 21)$  model which yields the normalized rate  $\frac{2}{3}$  for computation. The discussed approach generalizes to all network models of this class.

3) *The Case of  $(r+1) \bmod 3 = 2$* : For these networks, e.g.  $(4, 5)$ ,  $(7, 8)$ ,  $(10, 11)$ ,  $(13, 14)$ , vector network coding is again necessary over  $n = 3$  channel uses. The code for the “indecomposable”  $(4, 5)$  model is provided in Figure 4. For other models in this series, we repeat our reasoning. As an example, vector coding with  $n = 3$  for the model  $(7, 8)$  is equivalent to scalar linear coding for the  $(21, 24)$  model. The  $(21, 24)$  model is first “reset” by subtracting out 9 dimensions (achieving 6 computations), resulting in a  $(12, 15)$  model which is equivalent to coding for the  $(4, 5)$  model over  $n = 3$  channel uses. Similarly, all network models of this class are “reset” to yield the indecomposable  $(4, 5)$  model. ■

### B. Upper Bound for Linear Coding Capacity for Computation

*Lemma 3 (Upper Bound on Linear Coding Capacity):*

Consider an  $(m, L)$  network model with  $p = 2$  and linear encoding and decoding operations defined over  $\mathbb{F}_2$ . Let  $\alpha \triangleq \frac{m}{L}$ . In the regime  $0 \leq \alpha < 1$ ,

$$\frac{R_{COMP}^{lin}}{L} \leq \frac{2}{3}. \quad (5)$$

*Proof:* Consider any vector linear code over  $n$  channel uses for an arbitrary  $(m, L)$  network model. The vector linear code over the model  $(m, L)$  is equivalent to a scalar linear code over the model  $(nm, nL)$ . The linear encoding is characterized by a set of  $k$  beamforming vectors at both transmitters:  $\{\Phi_j\}_{j=1}^k$  and  $\{\Gamma_j\}_{j=1}^k$ . Assuming that  $k$  computations are successfully recovered by linear decoding at *both* receivers, our aim is to prove  $\frac{k}{nL} \leq \frac{2}{3}$ . The observed vectors at both receivers are

$$Y_1 = \sum_{j=1}^k a_j \Phi_j + \sum_{j=1}^k b_j G_\gamma \Gamma_j,$$

$$Y_2 = \sum_{j=1}^k a_j G_\gamma \Phi_j + \sum_{j=1}^k b_j \Gamma_j.$$

1) *Balancing Multi-Receiver Demands:* For  $1 \leq j \leq k$ , if the function  $a_j + b_j$  is to be recovered at *both* receivers, each receiver must either decode the computation  $a_j + b_j$  or decode  $a_j$  and  $b_j$  separately (and then compute  $a_j + b_j$  afterward). Consider the first receiver which observes the vector  $Y_1 \in \mathbb{F}_2^{nL}$  and attempts to decode the  $j$ -th computation  $a_j + b_j$ . In any linear code, either  $\Phi_j = G_\gamma \Gamma_j$  and  $a_j + b_j$  can be decoded together as one computation (taking up one dimension at the receiver), or  $\Phi_j \neq G_\gamma \Gamma_j$  which means  $a_j$  and  $b_j$  lie on two separate linearly independent vectors. Similarly, for the second receiver observing  $Y_2 \in \mathbb{F}_2^{nL}$ , either  $G_\gamma \Phi_j = \Gamma_j$  or  $G_\gamma \Phi_j \neq \Gamma_j$ . Due to the properties of the network model, conditions involving both receivers hold in the case that  $\alpha < 1$  strictly and  $G_\gamma$  is not full rank:

$$\Phi_j = G_\gamma \Gamma_j \Rightarrow G_\gamma \Phi_j \neq \Gamma_j,$$

$$G_\gamma \Phi_j = \Gamma_j \Rightarrow \Phi_j \neq G_\gamma \Gamma_j.$$

2) *Checkerboard Proof:* Based on the above arguments for linear coding over  $\mathbb{F}_2$ , only three joint configurations are possible for the following received vectors  $\{\Phi_j, G_\gamma \Gamma_j\}$  at the first receiver and  $\{G_\gamma \Phi_j, \Gamma_j\}$  at the second receiver. Let  $d_{j,1}, d_{j,2}$  represent the dimension of the subspaces spanned by the received vectors at the two receivers respectively due to the  $j$ -th encoding vectors. Either  $(d_{j,1}, d_{j,2}) = (1, 2)$ ,  $(d_{j,1}, d_{j,2}) = (2, 1)$ , or  $(d_{j,1}, d_{j,2}) = (2, 2)$ . In order for *both* receivers to recover  $a_j + b_j$ , due to the assumption of linearity and the need for linear independence of received vectors, the received vectors  $\{\Phi_j, G_\gamma \Gamma_j\}$  at the first receiver must not overlap with the prior subspaces, and the received vectors  $\{G_\gamma \Phi_j, \Gamma_j\}$  at the second receiver must not overlap with prior subspaces. Imagine an integer square lattice that is of size  $nL \times nL$  representing the maximum number of subspace dimensions at both receivers. From the origin, imagine making  $k$  moves on the board: either we take two steps to the right and one step up, or one step to the right and two steps up. By basic geometry, the maximum number of moves is bounded as  $3k \leq 2nL$  to remain on the square grid. ■

## VI. CONCLUSION

Computation alignment strategies were introduced for function multicasting in a multi-receiver sum-network. Vector and

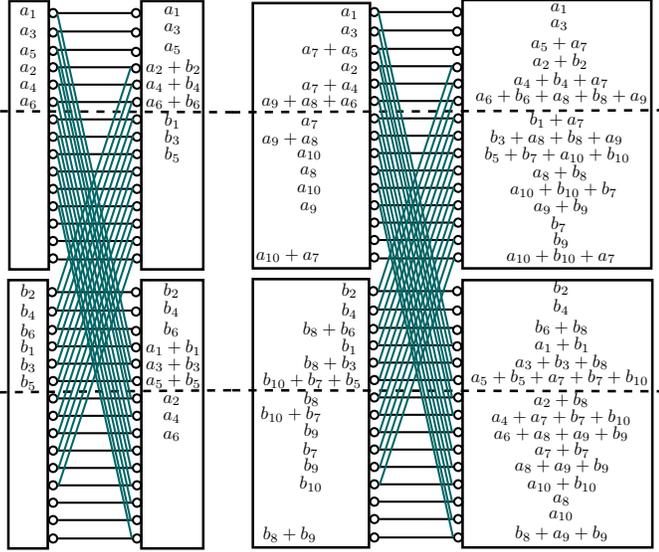


Fig. 4. Explicit scalar linear code for  $(m, L) = (12, 15)$ . The code corresponds to a *vector linear code* (over  $n = 3$  channel uses) of the  $(4, 5)$  “indecomposable” model. The computation rate achieved is optimal  $\frac{R_{comp}}{L} = \frac{10}{15} = \frac{2}{3}$ . On the left, a total of 6 computation bits are possible using alternating *computation alignment*. On the right, 4 more computation bits for both receivers are possible due to careful alignment of beamforming vectors within the received dimensions.

scalar linear coding strategies were analyzed. With multiple-receivers, a key challenge is to balance shared network resources with receiver demands. A good network code allows for *in-network* computation as opposed to recovering all messages at the receivers. The linear coding capacity for computation was determined for a countably infinite class of deterministic models. Several network equivalence and decomposition theorems were developed which have potential for the design of structured codes in multi-hop networks.

#### APPENDIX

Theorem 2 is due to Lemma 4 below which follows directly from interleaving arguments. The network equivalences apply to straight data transfer (any message configuration) as well as for function computation.

*Lemma 4 (Network Equivalence):*

- 1) For any  $n \in \mathbb{Z}$ ,  $(nm, nL) = (m, L) \times (m, L) \times \dots \times (m, L)$  (a total of  $n$  factors). As a byproduct of this fact, vector linear coding over  $n$  channel uses of an  $(m, L)$  network model is identical to scalar linear coding over the channel model  $(nm, nL)$ .
- 2)  $(2m + 1, 2L + 1) = (m, L) \times (m + 1, L + 1)$ .
- 3) “Gap-2” models. There are two cases, even and odd:
  - a) For any  $\ell \in \mathbb{Z}$ :  $(2\ell, 2\ell + 2) = (\ell, \ell + 1) \times (\ell, \ell + 1)$ . This is a special case of item 1.
  - b) For any  $\ell \in \mathbb{Z}$ :  $(2\ell + 1, 2\ell + 3) = (\ell, \ell + 1) \times (\ell + 1, \ell + 2)$  (special case of item 2)
- 4) “Gap-3” models. There are three cases:
  - a) For any  $\ell \in \mathbb{Z}$ :  $(3\ell, 3\ell + 3) = (\ell, \ell + 1) \times (\ell, \ell + 1) \times (\ell, \ell + 1)$  (special case of item 1)
  - b) For any  $\ell \in \mathbb{Z}$ :  $(3\ell + 1, 3\ell + 4) = (\ell, \ell + 1) \times (\ell, \ell + 1) \times (\ell + 1, \ell + 2)$

- c) For any  $\ell \in \mathbb{Z}$ :  $(3\ell + 2, 3\ell + 5) = (\ell, \ell + 1) \times (\ell + 1, \ell + 2) \times (\ell + 1, \ell + 2)$

Observe that the “Gap-3” models can be decomposed into three “Gap-1” models.

- 5) “Gap- $k$ ” models: In general, we can give the following formula:

$$(m, L) = (r, r + 1)^{L-m-a} \times (r + 1, r + 2)^a,$$

where

$$r = \left\lfloor \frac{m}{L - m} \right\rfloor \quad \text{and} \quad a = m \bmod (L - m)$$

*Proof Outline for the Last Item.* In the graph representation of the  $(m, L)$  model, color the vertices of the first user with  $L - m$  colors, starting from the top and periodically going through all colors. Use exactly the same coloring for the receiving vertices of the first user as well as for both transmitting and receiving vertices of the second user. Direct inspection reveals that each color represents an orthogonal component model, and that each component model is again a standard symmetric model, albeit of the form  $(r, r + 1)$ . A tedious but straightforward calculation leads to the claimed formula.

#### REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans. on Information Theory*, vol. 46, July 2000.
- [2] S.-Y. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Trans. on Information Theory*, vol. 49, pp. 371–381, Feb. 2003.
- [3] R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Trans. on Networking*, vol. 11, pp. 782–795, Oct. 2003.
- [4] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. on Information Theory*, vol. 52, pp. 4413–4430, Oct. 2006.
- [5] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, “Polynomial time algorithms for multicast network code construction,” *IEEE Tran. Info. Theory*, vol. 51, pp. 1973–1982, 2005.
- [6] R. Dougherty, C. Freiling, and K. Zeger, “Insufficiency of linear coding in network information flow,” *IEEE Transactions on Information Theory*, vol. 51, pp. 2745–2759, Aug. 2005.
- [7] A. Das, S. Vishwanath, S. Jafar, and A. Markopoulou, “Network coding for multiple unicasts: An interference alignment approach,” in *Proc. of IEEE Intern. Symp. on Information Theory*, pp. 1878–1882, June 2010.
- [8] B. Rai and B. Dey, “On network coding for sum-networks,” *IEEE Transactions on Information Theory*, vol. 58, pp. 50–63, Jan. 2012.
- [9] A. Ramamoorthy and M. Langberg, “Communicating the sum of sources over a network,” *CoRR*, vol. abs/1001.5319, 2010.
- [10] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, “Network coding for computing: Cut-set bounds,” *IEEE Transactions on Information Theory*, vol. 57, pp. 1015–1030, Feb. 2011.
- [11] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, “Linear coding for network computing,” in *Proc. IEEE Intern. Symp. on Information Theory*, pp. 2652–2655, Aug. 2011.
- [12] R. Appuswamy and M. Franceschetti, “Computing linear functions with linear network codes,” in *Proc. 48th Annual Allerton Conference on Communication, Control, and Computing*, pp. 29–35, Oct. 2010.
- [13] A. Avestimehr, S. Diggavi, and D. Tse, “Wireless network information flow: A deterministic approach,” *IEEE Transactions on Information Theory*, vol. 57, pp. 1872–1905, April 2011.
- [14] B. Nazer and M. Gastpar, “Compute-and-forward: Harnessing interference through structured codes,” *IEEE Transactions on Information Theory*, vol. 57, pp. 6463–6486, Oct. 2011.
- [15] U. Niesen, B. Nazer, and P. Whiting, “Computation alignment: Capacity approximation without noise accumulation,” *CoRR*, vol. abs/1108.6312, 2011.
- [16] A. E. Gamal and Y.-H. Kim, *Network Information Theory*. New York: Cambridge University Press, 2011.
- [17] C. Suh, N. Goela, and M. Gastpar, “Computation in multicast networks: Function alignment and converse theorems,” *CoRR*, vol. abs/1209.3358, October 2012.