Article:

Mortazawi Molu, Mehdi, Cumanan, Kanapathippillai orcid.org/0000-0002-9735-7019 and Burr, Alister Graham orcid.org/0000-0001-6435-3962 (2016) Low-Complexity Compute-and-Forward Techniques for Multi-source Multi-relay Networks. IEEE Communications Letters. pp. 926-929. ISSN 1089-7798

Low-Complexity Compute-and-Forward Techniques for Multi-Source Multi-Relay Networks

Mehdi M. Molu, Kanapathippillai Cumanan, Alister Burr

***Abstract*—Compute-and-Forward (C&F) relaying in a multi-source multi-relay network is studied in this paper and two novel algorithms are proposed, addressing choice of integer matrix, taking into account the effect of singularity. The first algorithm assumes that there is no cooperation between the nodes for choosing proper integer vectors in the relay nodes; this method is referred to as "blind C&F" and an algorithm is proposed which guarantees that each relay chooses the best integer vector that contains information from at least m source nodes. In the second algorithm that is described as "partially coordinated C&F", we assume partial cooperation between the relay nodes and propose to exchange a single variable with which the relays are sorted for transmission. The performance of the proposed algorithm is nearly equivalent with optimal relaying which requires significant overhead signalling.**

***Index Terms*—Compute-and-Forward, Computation Rate, Physical layer Network Coding**

I. Introduction

Compute-and-Forward (C&F) [1] is a relatively new relaying technique that relies on lattice codes through the linearity property of lattices. Exploiting this property of a lattice allows multiple source nodes in a network to transmit data simultaneously using the same resources (time and frequency). This method of relaying that is referred to as PLNC (Physical Layer Network Coding) in the literature is in contrast with conventional relaying methods (e.g., amplify-and-forward or decode-and-forward [2], [5]) wherein the source nodes transmit their data orthogonally using different resources. In the context of C&Frelaying in fading channels, one challenging task is to find corresponding integer vectors in the relay nodes which is referred to as a network coding vector or an $\vec{a}$ vector in the literature (e.g., [1]). There is intensive ongoing research on finding the best network coding vector ($\vec{a}$ vector), however, so far, most of the focus of the literature is on obtaining an integer vector $\vec{a}$ that maximizes the computation rate in the relay regardless of the corresponding $\vec{a}$ vectors computed in other relay nodes (i.e., local maximization). Moreover, one necessary condition under which a destination node is capable of unambiguously decoding transmitted information by the source nodes is that the matrix (say $\vec{A}$ matrix) obtained using the $\vec{a}$ vectors computed in the relay nodes must be non-singular. Using conventional methods for computing network coding vectors in the relays, although the rate is maximised locally, the overall computation rate of the network is zero if $\vec{A}$ matrix is singular (i.e., if $|\vec{A}| = 0$).

This work was supported in part by European Commission FP7 contract no. 318177 (DIWINE), and in part by EPSRC Grant no. EP/K040006 (NetCoM5G).

Authors are with the Department of Electronics, University of York, UK, email:firstname.lastname@york.ac.uk

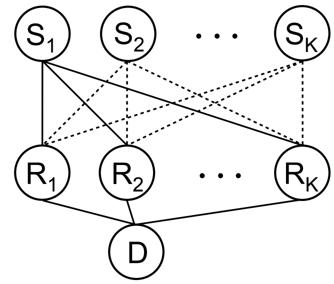

Fig. 1. System Model

There are few paper that directly address the problem of finding a proper $\vec{A}$ matrix that is not singular; in particular, [6], [7] study a similar problem to the one that we address in this paper, however, the problem is usually tackled assuming full coordination between the nodes that indeed imposes significant overhead signalling in practice.

Contribution: In this paper we study a multi-source multi-relay network and aim to compute network coding vector $\vec{a}$ in the relay nodes that tries to avoid singular $\vec{A}$. Based on two different assumptions, we propose two new algorithms in this paper:

- Non-coordinated (blind) C&F where the relay nodes compute network coding vectors blindly, without knowledge of the network coding vectors used in the other relay nodes.
- Partially coordinated C&F where the relay nodes partially communicate to specify the order of transmission and use a network coding function that does not reduce the rank of the $\vec{A}$ matrix.

The paper is organized as follows: In Section II system model is introduced and the rate description of C&Frelaying is provided. In Section III two novel relaying strategies are proposed. In Section IV numerical simulations are provided to validate the usefulness of the proposed methods and the paper is finalized by some concluding remarks in Section V.

II. System Model

As shown by Fig. 1, a cooperative network consisting of K source nodes, K relay nodes and one destination node is studied. The entire transmission from sources to the destination is divided into $K+1$ time slots: in the first time slot all the source nodes transmit their data to the destination using a shared interference channel. In a second phase, that consists of K time slots, the relay nodes each compute an equation from the received superimposed signal and forward it to the destination node. The relay nodes exploit C&F and so in the second phase, relay nodes use orthogonal channels for transmission because the source node requires at least K equations to be capable of decoding all the messages transmitted from the source nodes.The transmissions from the source and relay nodes are summarized in the following:

Source: Each source node selects a message $\vec{w}_l$ that is drawn from a set of M messages with equal probability. Every message is then mapped to a nested lattice codeword $\vec{x}_l$ and sent to the relay nodes in the first time slot.

Relay: Since the relay nodes exploit C&Frelying, each relay exploits lattice decoding and attempts to find a set of equations and the rates corresponding to each particular equation. Let us define the set $\mathbb{A}_r$ as a set of integer vectors defining possible network coding functions at relay r as follows:

$$\mathbb{A}_r = \{\vec{a}_{r,1}, \vec{a}_{r,2}, \cdots, \vec{a}_{r,n}\}. \tag{1}$$

Each integer vector $\vec{a}_{r,i}$ results in a computation rate that is stored in set $\mathbb{R}_r$ as follows:

$$\mathbb{R}_r = \{\mathcal{R}_{r,1}, \mathcal{R}_{r,2}, \cdots, \mathcal{R}_{r,n}\}. \tag{2}$$

It is assumed that $\mathcal{R}_{r,1} \geq \mathcal{R}_{r,2} \geq \cdots \geq \mathcal{R}_{r,n}$. The relay function will be discussed in further detail in the next sections, however, note that it is proved in [1] that the computation rate in relay node r is obtained using following expression:

$$\mathcal{R}_r(\vec{h}, \vec{a}) = \log_2^+ \left(\frac{\gamma}{\gamma \parallel \alpha_{r,l}\vec{h}_r - \vec{a}_{r,l} \parallel^2 + \alpha_{r,l}^2} \right) \tag{3}$$

which depends on signal-to-noise ratio γ, inflation coefficient $\alpha_{r,l}$, channel realisation $\vec{h}_r$ and choice of the integer vector $\vec{a}_{r,l}$ (see [1] for detailed description of the parameters).

It is clear that choosing $\vec{a}_{r,1}$ is the best option if the intention is to maximize the computation rate locally in the relay nodes; this is indeed the main optimization criterion in the original C&Fpaper in [1]. However, in this paper, we are interested in optimizing the overall transmission rate of the network, defined as follows:

$$\mathcal{R}(\vec{H}, \vec{A}) = \begin{cases} \min(\mathcal{R}_1, \cdots, \mathcal{R}_K), & \text{if } |\vec{A}| \neq 0 \\ 0, & \text{if } |\vec{A}| = 0 \end{cases} \tag{4}$$

where $\vec{H}$ is the channel realization between the source and the relay nodes and $\vec{A}$ is the matrix whose columns are the $\vec{a}_r$ vectors exploited in the relay nodes as the network coding vectors. We assume that each relay appends its chosen integer vector to the equation and transmits it to the destination; also we assume that the relays can overhear one another's transmissions.

III. Relay Strategy

Upon reception of the source transmissions, each relay node r needs to choose an integer coefficient $\vec{a}_r$ and perform lattice decoding before forwarding an equation towards the destination. One can assume different criteria for computing $\vec{a}_r$ vectors as described in following subsections.

A. Non Coordinated (Blind) Compute-and-Forward

Once the destination collects the relay transmissions, it will be capable of decoding the messages from the source nodes if the matrix $\vec{A}$ is a full rank matrix. In an attempt to reduce the occasions which result in non full rank $\vec{A}$, a blind C&Frelay strategy is proposed in the following:

Proposition: Instead of computing an equation that corresponds to the highest computation rate in the relay r (i.e., locally optimizing rate), each relay computes a set of equations corresponding to different computation rates as described in (1) and (2). Moreover a new parameter is defined as

$$\kappa_r = \{k_{r,1}, k_{r,2}, \cdots, k_{r,n}\} \tag{5}$$

which specifies the number of non-zero entries in $\vec{a}_r$ vectors. As an example, an integer vector $\vec{a}_{r,j} = [1, 0, 0]$ consists of information only from source 1, however an integer vector $\vec{a}_{r,j} = [1, 0, 1]$ consists of information from two source nodes, source 1 and source 3. We define $k_{r,j}$ as the number of non-zero entries in the $\vec{a}_{r,j}$ vector; i.e., $k_{r,j} = \mathrm{nnz}(\vec{a}_{r,j})$. As a relaying strategy, instead of forwarding $\vec{a}_{r,1}$, we propose to transmit a function that includes information from, at least, m sources, i.e., $\vec{a}_r = \vec{a}_{r,j}$ where

$$\mathcal{R}_{r,j} = \max\{\mathbb{R}_r\} \text{ given } k_{r,j} \geq m. \tag{6}$$

This strategy is helpful, especially at low SNR where the integer vectors $\vec{a}_{r,j}$ usually have only one non-zero entry; therefore, once the integer vector from other relays has a non-zero entry at the same position of $\vec{a}_{r,j}$, $|\vec{A}|$ becomes equal to zero, hence, setting the overall transmission rate of the network to zero; whereas, ensuring that at least m entries of the integer vectors $\vec{a}_{r,j}$ are non-zero, the probability of non-full rank $\vec{A}$ matrix decreases, hence avoiding $\mathcal{R}(\vec{H}, \vec{A}) = 0$ due to $|\vec{A}| = 0$. In Section IV computer simulations are provided to validate the benefits of the proposed algorithm.

B. Partially Coordinated Compute-and-Forward

In Blind C&Falgorithms, the relay nodes are indexed arbitrarily and so there is no rule to decide the order with which the relays transmit their equations. In other words, it is implicitly assumed that relay R_1 transmits first, and then the relay R_2 and etc. However, for a partially coordinated C&Falgorithm as proposed in this section we define a parameter referred to as *rate-difference* as follows:

$$d_r = \mathcal{R}_{r,1} - \mathcal{R}_{r,2}, \tag{7}$$

that is the rate-difference between two largest rates in each relay. In the following, it will be proposed to give the priority for transmission to the relay nodes with larger d_r; for instance, in a two relay scenario, if $d_2 > d_1$, the relay R_2 transmits its computed equation first and then the relay R_1 transmits an equation.

Proposition: Partially coordinated C&Fprotocol proposed in this section consists of two parts: i) sorting relays and specifying the priority of the transmission and ii) choosing the best equation in the relays (i.e., choosing proper integer vector $\vec{a}$) which simultaneously guarantees local optimization

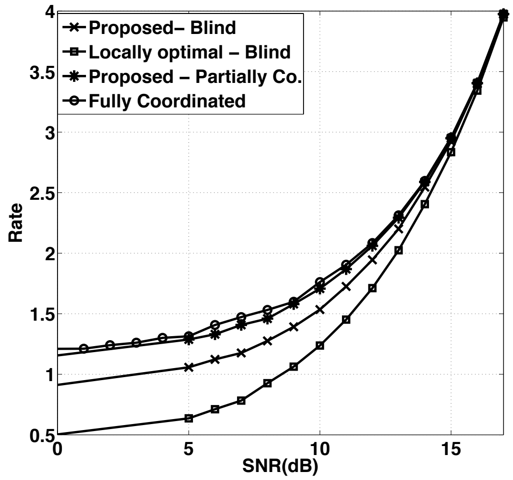

Fig. 2. Computation rate: 3 user ($K = 3$).

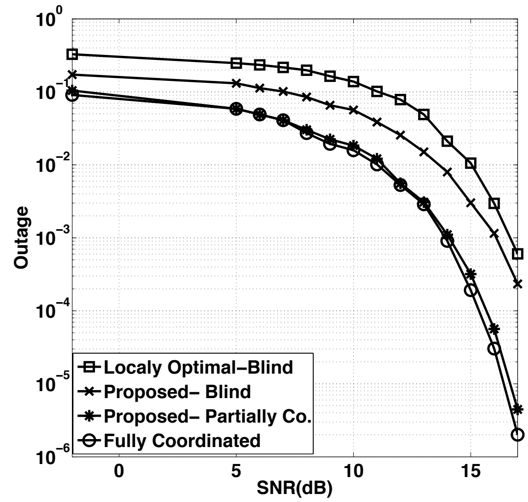

Fig. 3. Outage rate: 3 user ($K = 3$) and threshold rate $\mathcal{R}_{th} = 1$.

of the computation rate as well as preserving the rank of the $\vec{A}$ matrix. The algorithm is described in the following:

- Upon reception, every relay computes a set of best equations, leading to largest rates and corresponding rates with which the relays calculate the rate-difference and broadcast it. Since we assume the relays can overhear each other, each relay receives the rate-difference of other relays and based on the rate-differences, the relays are ordered for transmission as described earlier; i.e., the relays with larger rate-difference d_r get priority for transmission. The motivation for this is described throughout this section.
- For simplicity of notation, let us assume that the relay indices specify the order of transmission. In other words, we assume that $d_1 > d_2 > \cdots > d_n$ and so, R_1 is the first relay to transmit an equation, R_2 is the second relay and similarly, R_n is the last relay that transmits. Each relay appends the exploited integer vector to the frame and sends it to the destination. For instance, R_1 sends its integer vector $\vec{a}_1$ along with the equation; the R_2 overhears the $\vec{a}_1$ and exploits an integer vector $\vec{a}_2$ that does not reduce the rank of $[\vec{a}_1; \vec{a}_2]$ matrix. Relay R_3 overhears and decodes $\vec{a}_1$ and $\vec{a}_2$ from R_1 and R_2 transmissions and exploits a proper $\vec{a}_3$ that does not reduce the rank of $[\vec{a}_1; \vec{a}_2; \vec{a}_3]$. The transmission continues until all the relays transmit their corresponding data while ensuring that choosing an integer vector $\vec{a}_r$ does not lead to a non full-rank $\vec{A}$ matrix.

In order to better understand the algorithm, an example is provided in the following.

Example: Assume a network with three source and three relay nodes, operating at SNR= 10 dB, with channel realizations between the source and the relay nodes as follows:

$$\begin{aligned} \vec{h}_1 &= [0.85, 3.63, 1.91]^T \\ \vec{h}_2 &= [0.14, 13.7, 7.52]^T \\ \vec{h}_3 &= [2.37, 0.92, 4.51]^T . \end{aligned} \tag{8}$$

For each relay, one can compute a set of integer vectors $\mathbb{A}_r = \{\vec{a}_{r,1}, \vec{a}_{r,2}, \vec{a}_{r,3}, \cdots\}$ with which the rates of $\mathbb{R}_r = \{\mathcal{R}_{r,1}, \mathcal{R}_{r,2}, \mathcal{R}_{r,3}, \cdots\}$ can be achieved (note that we assume the entries of $\mathbb{R}_r$ are ordered in descending order). For instance for relay R_1, we have computed $\mathbb{A}_1$ and $\mathbb{R}_1$, with two entries, as follows:

$$\mathbb{A}_1 = \{\vec{a}_{1,1}, \vec{a}_{1,2}\} \quad \text{and} \quad \mathbb{R}_1 = \{\mathcal{R}_{1,1}, \mathcal{R}_{1,2}\}$$

where

$$\vec{a}_{1,1} = [0, 0, 1], \ \vec{a}_{1,2} = [1, 2, 5] \tag{9}$$

$$\mathcal{R}_{1,1} = 0.971, \quad \mathcal{R}_{1,2} = 0.943 \tag{10}$$

Likewise, one can compute the entries of $\mathbb{A}_2$ and $\mathbb{A}_3$ as follows:

$$\vec{a}_{2,1} = [0, 0, 1], \ \vec{a}_{2,2} = [2, -5, 5] \tag{11}$$

$$\vec{a}_{3,1} = [1, 0, 0], \ \vec{a}_{3,2} = [0, 0, 1] \tag{12}$$

and $\mathbb{R}_2$ and $\mathbb{R}_3$ as follows

$$\mathbb{R}_2 = \{3.15, 1.77\} \quad \text{and} \quad \mathbb{R}_3 = \{1.15, 1.01\}.$$

Consequently, the rate-difference d_r defined in (7) for the three relays can be defined as

$$d_1 = 0.028, \ \ d_2 = 1.38 \ \text{ and } \ d_3 = 0.14. \tag{13}$$

Since $d_2 > d3 > d1$ in (13), we propose to order relay transmission based on the rate loss. In this example, second relay R_2 transmits as the first relay because the largest rate loss occurs in R_2; therefore, it selects the best $\vec{a}$ vector corresponding to largest rate; i.e., the second relay chooses $\vec{a}_2 = [0, 0, 1]$ that corresponds to $\mathcal{R}_2 = 3.15$. Along with the transmission of the equation based on $\vec{a}_2$, the relays transmit $\vec{a}$ vector too. Upon reception of the $\vec{a}$ vector by the other relays, they decode it and store for future use. Now there are two more relays to transmit their equations, however, since $d_3 > d_1$, the third relay transmits first. The best option for third relay is to choose $\vec{a}_3 = [1, 0, 0]$ and note that this choice does not reduce the rank of $\vec{A}$ matrix. Relay R_3 sends its equation

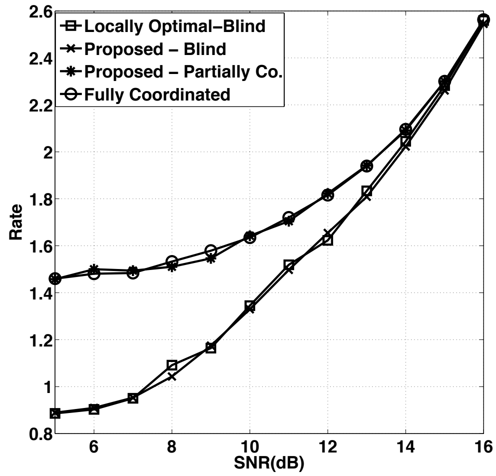

Fig. 4. Computation rate: 5 user ($K = 5$).

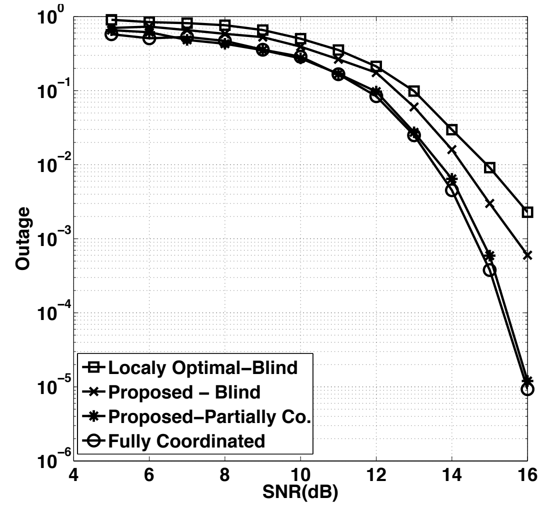

Fig. 5. Outage rate: 5 user ($K = 5$) and threshold rate $\mathcal{R}_{th} = 1.5$.

along with the chosen $\vec{a}_3$ that is overheard and decoded by relay R_1. The first relay is the last relay to send its equation, however, although the best option for relay R_1 is $\vec{a}_1 = [0, 0, 1]$, this choice reduces the rank of $\vec{A}$ matrix and sets $|\vec{A}| = 0$; therefore it selects second integer vector from set $\mathbb{A}_1$, i.e., $\vec{a}_1 = [1, 2, 5]$. Note that although the first relay selects its second best $\vec{a}$ vector, it leads to insignificant rate loss because the corresponding rate-difference is low ($d_1 = 0.028$). This is indeed the main motivation for the partially coordinated C&F. Note that if an integer vector corresponding to the best rate in a relay lead to a singular $\vec{A}$, the relay selects another integer vector with lower rate but full rank $\vec{A}$. Therefore, if a relay with larger d_r is forced to choose its second integer vector, this will lead to large rate loss in the relay nodes locally, and so we propose to give priority for transmission for the relays with larger d_r.

IV. Numerical Results

In this section numerical results for two relay networks with three and five source/relay nodes are provided (i.e., $K = 3$ and $K = 5$ in Fig. 1). In Fig. 1 we assume that the distance between any two neighbouring nodes is one meter and the path loss coefficient is $\alpha = 3$. We assume block Rayleigh fading channels that are obtained through $h_{ij} = (\frac{d_{ij}}{d_0})^{-\alpha}\tilde{h}_{ij}$ where $\tilde{h}_{ij}$ represent fading realisation between S_i and R_j. d_0 is the largest distance between a source node and a relay node.

Fig. 2 illustrates the computation rate (defined in (4)) using the proposed blind C&Falgorithm; each relay makes sure that network coding function includes data from at least two transmitters, i.e. $k_{r,j} \geq 2$ in (6). For comparison, the computation rate of the conventional blind C&Falgorithm is also provided; it is clear that the proposed blind C&Fachieves higher rates. Fig. 3 illustrates the outage rate assuming threshold rate $\mathcal{R}_{th} = 1.5$. Clearly, the outage rate of the proposed blind algorithm is lower than that of the conventional blind algorithm; this validates the usefulness of the blind C&Falgorithm proposed in this paper. In Fig. 2 and 3, the computation rate and outage are also shown for the partially coordinated C&Falgorithm. It is clear that the proposed algorithm that is developed by exchanging a few parameters among the relays approaches the fully coordinated C&Fthat requires significant signalling. Fig. 4 and Fig. 5 illustrate computation rate and outage for a system with five source and relay nodes. The superior performance of the proposed algorithms is evident.

V. Conclusion

Compute-and-Forward (C&F) relaying in a multi-source multi-relay network is studied in this paper and two novel algorithms are proposed. Assuming no coordination between the nodes, a blind C&Ftechnique is developed. Another algorithm is proposed that requires the exchange of a few parameters between the nodes. This algorithm is called partially coordinated C&Fand it is demonstrated to perform nearly as well as a fully coordinated C&Fsystem.

References

[1] 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.

[2] M. Molu and N. Goertz, “Optimal precoding in the relay and the optimality of largest eigenmode relaying with statistical channel state information,” *IEEE Transactions on Wireless Communications*, vol. 13, pp. 2113–2123, Apr. 2014.

[3] K. Cumanan, Y. Rahulamathavan, S. Lambotharan, and Z. Ding, “Mmse-based beamforming techniques for relay broadcast channels,” *IEEE Transactions on Vehicular Technology*, vol. 62, pp. 4045–4051, Oct 2013.

[4] L. Wei and W. Chen, “Compute-and-forward network coding design over multi-source multi-relay channels,” *IEEE Transactions on Wireless Communications*, vol. 11, pp. 3348–3357, Sep. 2012.

[5] Z. Chen, P. Fan, and K. Ben Letaief, “Compute-and-forward: Optimization over multisource-multirelay networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 1806–1818, May 2015.