Variable-Rate Network Coding for Multi-Source Cooperation

Rong Zhang and Lajos Hanzo
School of ECS, Univ. of Southampton, SO17 1BJ, UK.
Tel: +44-23-80-593 125, Fax: +44-23-80-593 045
Email: rz,lh@ecs.soton.ac.uk, http://www-mobile.ecs.soton.ac.uk

***Abstract*— Multi-Source Cooperation (MSC) techniques, including conventional Code Division Multiplexing (CDM) and Classic Network Coding (CNC) are investigated. We adopt a soft sum-product decoding algorithm for the CNC technique and propose a flexible Variable-rate Network Coding (VNC) technique. The iterative decoding convergence of the multiple source computation methods is analysed with the aid of EXtrinsic Information Transfer (EXIT) charts.**

I. Introduction

A fundamental issue of the Multiple Source Multiple Relay (MSMR) cooperation is the processing of numerous source information streams during their relaying [1]. In this paper, we focus our attention on this issue in the context of Multi-Source Cooperation (MSC) [2]–[4], which constitutes a specific instantiation of the MSMR scenario. MSC was first considered in [2], where the authors proposed the so-called parity-check based MSC regime using Read-Solomon (RS) codes to jointly encode the multiple information streams at the relay. The authors of [3] further proposed the Complex Field Coding (CFC) aided MSC, where the multiple source information streams are combined in order to yield values in the complex field. To enhance the channel coding gain attainable in fading channels, we proposed a high throughput MSC framework [5] and extended it to a multiplexed coding regime with the aid of a Low Density Generator Matrix (LDGM) based design [4].

Apart from the superposition-type coding of [3], [5] and joint channel encoding of [2], [4], the performance of the Classic Network Coding (CNC) scheme [6] has not been explored in the context of MSC. Firstly, the decoding of a particular information stream x_a from the composite CNC stream of $x_a \oplus x_b \oplus x_c \oplus \dots$ requires the knowledge of the correct information of all the rest of the network's information streams. However, this knowledge is not readily available in MSC at the destination. Furthermore, the mapping ambiguity of CNC, which is represented by $x_a \oplus x_b \oplus x_c \oplus \dots = x_c \oplus x_b \oplus x_a \oplus \dots$, prevents its employment in MSC, which was also stated in [3]. To overcome this problem:

- *we consider a range of multiple source processing techniques, from the CDM concept to the CNC technique.*
- *we conceive the sum-product decoding of CNC for efficiently processing a large number of information streams.*
- *we propose a novel near capacity Variable-rate Network Coding (VNC) regime without necessitating a sophisticated joint channel and network code design.*

Acknowledgements: The financial support of the EPSRC under the auspices of the UK-India Advanced Technology Centre in Wireless Communications and that of the EU's OPTIMIX project is gratefully acknowledged.

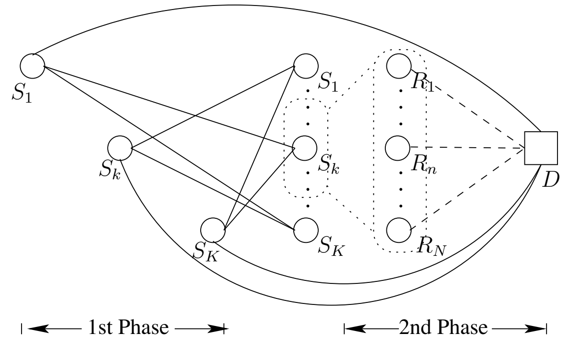


Fig. 1. Schematic of K sources cooperation employing TDD.

We organise our paper as follows. In Section II, we describe the MSC regime considered and introduce a range of multiple source stream processing methods with their iterative receiver algorithms, while our EXIT chart analysis is detailed in Section III. Finally, we present our simulation results in Section IV and conclude in Section V.

II. System Description

A. Cooperation Scenario

1) Overview: Consider a cooperation scenario involving K sources and a single destination D, where these K sources form a cluster for cooperatively transmitting their information to the destination. Our MSC regime is constituted by two cooperative phases using Time Division Duplexing. We assume furthermore that both the K sources and the destination employ a single antenna and all transmissions are synchronised.

In the first transmission phase, each of the K sources sends its information within the K orthogonal channels constituted by K non-overlapping time slots to the destination as well as to the rest of the sources that are listening, as seen in Fig 1. In the second transmission phase, $N \in (1, K]$ sources form a relay cluster, where each of the N relays exploits all the information available in the cluster *distributively* and transmits the appropriately processed information of the K sources to the destination *simultaneously* with the transmissions of all other relays in a non-orthogonal manner for the sake of achieving a high cooperation efficiency, as seen in the multiple access channel model of Fig 1. Finally, the information received during the two phases is jointly processed at the destination.

2) First Phase Cooperation: We commence by considering the first phase of the kth source S_k, where the information bit vector of length N_b is denoted by $\mathbf{b}_k^I$. Following channel coding and modulation, we generate the transmit data vector $\mathbf{x}_k^I$ of length N_x. Thus, the discrete time baseband system's first phase model of source S_k describing its transmission to the destination as well as to relay n are given by:

$$\mathbf{y}_D^I = h_{k,D}^I \mathbf{x}_k^I + \mathbf{n}_{k,D}^I, \quad \mathbf{y}_n^I = h_{k,n}^I \mathbf{x}_k^I + \mathbf{n}_{k,n}^I, \quad k \neq n \quad (1)$$

where $\mathbf{y}_D^I$ and $\mathbf{y}_n^I$ denote the received signal vector of length N_x at the destination and at the nth relay, respectively. Further, $h_{k,D}^I$ and $h_{k,n}^I$ denote the quasi-static flat Rayleigh fading channel between source k and the destination as well as between source k and relay n, respectively. Finally, $\mathbf{n}_{k,D}^I$ and $\mathbf{n}_{k,n}^I$ denote the complex-valued Additive White Gaussian Noise (AWGN) vector of length N_x.

The signal $\mathbf{y}_D^I$ received at the destination will be jointly processed in conjunction with the signal $\mathbf{y}_D^{II}$ received at the destination during the second phase. On the other hand, the signal $\mathbf{y}_n^I$ received at relay n is first decoded and then prepared for the second transmission phase[1].

3) Second Phase Cooperation: Relay n has access to all the information in the cluster that may be collectively represented as $\{\tilde{\mathbf{b}}_{n,1}^I, \ldots, \tilde{\mathbf{b}}_{n,K}^I\}$, where $\tilde{}$ denotes the estimated version of the parameter, which is potentially erroneous, with the exception of $\tilde{\mathbf{b}}_{n,n}^I = \mathbf{b}_{n,n}^I$. It then performs different MSC operations and generates the resultant transmit data vector as $\mathbf{x}_n^{II}$ of length N_x. Hence the discrete time baseband system model of the concurrent second phase transmissions of the N relays to the destination is formulated as:

$$\mathbf{y}_D^{II} = \sum_{n=1}^{N} h_{n,D}^{II} \mathbf{x}_n^{II} + \mathbf{n}_D^{II}, \quad (2)$$

where $\mathbf{y}_D^{II}$ denotes the received signal vector of length N_x at the destination and $h_{n,D}^{II}$ denotes the channel between relay n and the destination, which is also modelled by quasi-static flat Rayleigh fading. Furthermore, $\mathbf{n}_D^{II}$ denotes the complex-valued AWGN vector of length N_x[2].

B. Computation Methods At Relay

Let us first define the normalised throughput per relay node as $\eta = Krb/W$, where r is the channel coding rate, $b = 1$ is the number of bits per symbol assuming BPSK modulation, W is the total number of orthogonal channels required to convey K sources' information by the relay node.

1) Code Division Multiplexing: Non-orthogonal CDM refers to the transmission of K bit vectors using K unique non-orthogonal *codes*, where the codes may be constructed by a *source-specific* channel code or interleaver. At relay n, each of the K bit vectors $\tilde{\mathbf{b}}_{n,k}^I$ is first channel coded by a stream-specific coding function $f_{c,k}$ to yield a length-N_c coded bit vector $\mathbf{c}_{n,k}^{II}$ and then BPSK modulated, as characterized by the function f_m, in order to yield a length-N_c modulated data vector $\mathbf{x}_{n,k}^{II}$. Finally, the resultant K modulated data vectors are superimposed at relay n in order to form the composite transmit data vector $\mathbf{x}_n^{II}$ of length N_c:

$$\mathbf{x}_n^{II} = \sum_{\forall k} \rho_k \mathbf{x}_{n,k}^{II} = \sum_{\forall k} \rho_k f_m[f_{c,k}(\tilde{\mathbf{b}}_{n,k}^I)], \quad (3)$$

where $\rho_k = \sqrt{P_n/K}$ is the equally shared transmit power of the kth non-orthogonal code-based channel. Furthermore, we opted for employing stream-specific random interleavers to ensure the unique distinction of the K superimposed layers. Hence, the normalised throughput per relay node in the CDM scenario considered is $\eta_{cdm} = Kr$.

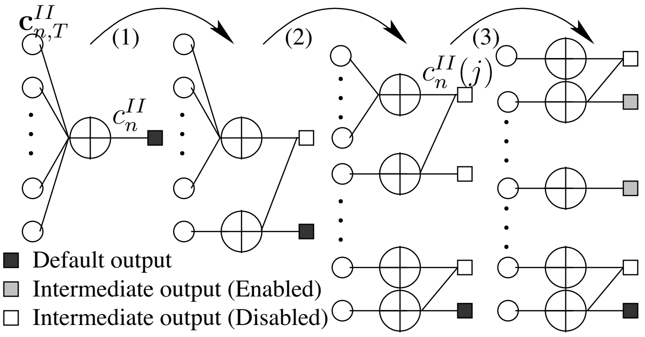


Fig. 2. Factor graph of packet-wise r_∞-VNC (left), the transition graphs (middle) and the ultimately transformed graph of general VNC with two enabled intermediate outputs.

2) Element-wise CNC: In the CNC scheme, each of the K information bit vectors is first subjected to stream-specific channel coding at relay n, yielding the coded bit vector $\mathbf{c}_{n,k}^{II}$ of length N_c. These K coded bit vectors are then element-wise 'XOR'ed to yield a composite CNC bit vector $\mathbf{c}_n^{II}$ of length N_c. Finally, the CNC bit vector $\mathbf{c}_n^{II}$ is BPSK modulated by the function f_m in order to form the length-N_c transmit data vector $\mathbf{x}_n^{II}$. Thus, we have:

$$\mathbf{x}_n^{II} = f_m[\mathbf{c}_n^{II}] = f_m[\bigoplus_{\forall k} \mathbf{c}_{n,k}^{II}] = f_m[\bigoplus_{\forall k} f_{c,k}(\tilde{\mathbf{b}}_{n,k}^I)]. \quad (4)$$

As a result, CNC is the method that conceals the original source information, while the CDM is the method that retains the original information format. Note, the normalised throughput per relay node of the CNC scheme is also $\eta_{cnc} = Kr$.

Since CNC may be viewed as the element-wise parity check of the K coded bit vectors, the 'XOR' operation may be interpreted as the action of a $d_c = K$-degree check node. We thus explore the possibility of increasing the check node degree leading to our proposed packet-wise VNC scheme.

3) Packet-wise Bit-level VNC: At relay n, the K stream-specific coded bit vectors $\mathbf{c}_{n,k}^{II}, \forall k$ are Parallel to Serial (P/S) converted to yield a concatenated coded bit vector $\mathbf{c}_{n,T}^{II}$ of length KN_c, obeying $c_{n,T}^{II}(i) = c_{n,a_1}^{II}(a_2)$ with $a_1 = [(i - 1) \oplus K] + 1$ and $a_2 = \lceil i/K \rceil$, where $\lceil \cdot \rceil$ represents the largest integer operator. Then the 'XOR' operation is performed for all the parallel bits of $\mathbf{c}_{n,T}^{II}$ (packet-wise) in order to yield a single bit c_n^{II}. This is then followed by the classic BPSK modulation, as described by the function f_m. Hence, we have

$$x_n^{II} = f_m(c_n^{II}) = f_m[\bigoplus_{i=1}^{KN_c} c_{n,T}^{II}(i)]. \quad (5)$$

[1] *Remarks 1* There are numerous solutions that deal with the decoding errors of the first phase Decode-and-Forward (DF) at the relay. However, we focus our attention exclusively on the MSC techniques used at the relays and set aside the issue of decoding error counter-measures for future research.

[2] *Remarks 2* Let P^I and P^{II} denote the total transmission power available for the first and second phase, respectively and we let $P^I = P^{II} = P/2$, where P is the total available power that is normalised to unity. Furthermore, $P_n = P^{II}/N$ denotes the equally shared transmit power of relay n.

The left subplot of Fig 2 portrays the operation in Eq (5), where the KN_c bit inputs (variable nodes) of $\mathbf{c}_{n,T}^{II}$ denoted by circles are subjected to CNC checked (check node), which are denoted by the crossed-circle symbol in order to yield a single-bit output c_n^{II} denoted by the filled square symbol.

To elaborate further, sending only the CNC checked bit results into an extremely compressed packet. Insightfully, we may transform however the resultant single (KN_c+1)-edge check node to KN_c three-edge check nodes, as seen in stages (1), (2) and (3) of Fig 2. This transformation allows us to transmit any number of additional intermediate outputs, so that the extremely 'over-compressed' single-bit source packet is better separated, as seen in the right subplot of Fig 2, where two more outputs are seen to be enabled. Mathematically, the jth intermediate output may be written as:

$$c_n^{II}(j) = \bigoplus_{i=1}^{j} c_{n,T}^{II}(i) \qquad j \in [1, KN_c]. \tag{6}$$

Let Ω host the indices associated with the enabled outputs, the transmit data vector becomes $\mathbf{x}_n^{II} = f_m[c_n^{II}(j)], \forall j \in \Omega$.

By appropriately controlling the number of intermediate outputs, a conveniently controlled variable rate scheme arises. We thus refer to our proposed scheme as a VNC arrangement having a rate of r_v, where we refer to the particular arrangement transmitting the single CNC checked bit as r_∞-VNC and that transmitting all intermediate outputs as r_1-VNC. Note that the normalised throughput per relay node is $\eta_{vnc} = r_v r$.

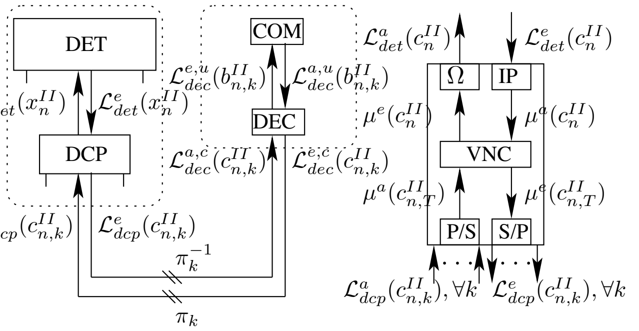
C. Iterative Receiver At Destination

1) Receiver Structure: Given the received signal observations of $\mathbf{y}_D^I, \mathbf{y}_D^{II}$ and the knowledge of CSI, we employ iterative receiver to pursue the Maximum *A posteriori* Probability (MAP) estimate of each source's information owing to the otherwise excessive complexity. Exchanging extrinsic information by iterating between the two receiver phases may be mutually helpful, but our focus is on the encoding and decoding techniques of the second transmission phase, hence we assume that the first phase source to destination transmission provides the second phase with a constant level of *a priori* mutual information of the multiple sources.

The iterative receiver of Fig 3 exchanges Log-Likelihood Ratios (LLR) of four building blocks. The first stage is the multiple relay based detection (DET), which separates the signals received from N relays. This is followed by the decomposition (DCP) of the K sources' information at each relay, where we collectively treat the decoding of each MSC technique as an instantiation of DCP. Each of the K decomposed information streams is then forwarded to the soft channel decoder (DEC) of Fig 3 associated with each of the K sources. Finally, combining (COM) of the N relays' signal is performed in order to achieve an improved decision.

2) Multiple Relay Detection: In this paper, we employ a low-complexity interference cancellation algorithm [7]. When considering x_n^{II} for example, the detector treats the interference contribution of $\mathbf{x}_{-n}^{II}$ as Gaussian noise, i.e. we have:

$$y_D^{II} = h_{n,D}^{II} x_n^{II} + \sum_{\forall m \neq n} h_{m,D}^{II} x_m^{II} + n_D^{II}. \tag{7}$$



DET
COM
$\mathcal{L}_{det}^a(c_n^{II})$
$\mathcal{L}_{det}^e(c_n^{II})$
$\mathcal{L}_{dec}^{e,u}(b_{n,k}^{II})$
$\mathcal{L}_{dec}^{a,u}(b_{n,k}^{II})$
Ω
IP
$\mathcal{L}_{det}^a(x_n^{II})$
$\mathcal{L}_{det}^e(x_n^{II})$
DEC
$\mu^e(c_n^{II})$
$\mu^a(c_n^{II})$
DCP
$\mathcal{L}_{dec}^{a,c}(c_{n,k}^{II})$
$\mathcal{L}_{dec}^{e,c}(c_{n,k}^{II})$
VNC
$\mu^a(c_{n,T}^{II})$
$\mu^e(c_{n,T}^{II})$
$\mathcal{L}_{dcp}^a(c_{n,k}^{II})$
$\mathcal{L}_{dcp}^e(c_{n,k}^{II})$
π_k^{-1}
P/S
S/P
π_k
$\mathcal{L}_{dcp}^a(c_{n,k}^{II}), \forall k$
$\mathcal{L}_{dcp}^e(c_{n,k}^{II}), \forall k$

Fig. 3. Iterative receiver of second phase cooperation with the detailed block of DCP for VNC method is shown on right.

As a result, the detector generates the *extrinsic* information of x_n^{II} after interference cancellation as:

$$\mathcal{L}_{det}^e(x_n^{II}) = \ln \frac{\exp(|y_D^{II} - M_\xi - h_{n,D}^{II}|^2/2V_\xi)}{\exp(|y_D^{II} - M_\xi + h_{n,D}^{II}|^2/2V_\xi)} \tag{8}$$

where M_ξ represents the mean of the interference term $\xi = \sum_{\forall m \neq n} h_{m,D}^{II} x_m^{II} + n_D^{II}$, while V_ξ denotes the variance of ξ:

$$M_\xi = \sum_{\forall m \neq n} h_{m,D}^{II} \tilde{x}_m^{II}, \quad V_\xi = \sum_{\forall m \neq n} v_m^2 |h_{m,D}^{II}|^2 + (\sigma_D^{II})^2,$$

where σ_D^{II} is the noise standard deviation for the second transmission phase. The soft estimate $\tilde{x}_m^{II} = \tanh[\mathcal{L}_{det}^a(x_m^{II})/2]$ and its instantaneous variance $v_m = 1 - (\tilde{x}_m^{II})^2$.

3) Multiple Source Decomposition: The decomposition operation required for CDM may be combined with the DET stage. By substituting Eq (3) into Eq (2), we arrive at:

$$\mathbf{y}_D^{II} = \sum_{\forall n} h_{n,D}^{II} \sum_{\forall k} \rho_k \mathbf{x}_{n,k}^{II} + \mathbf{n}_D^{II}. \tag{9}$$

Hence, we may carry out the DCP and DET operations of Fig 3 in a joint manner in order to get the *extrinsic* information of $\mathcal{L}_{dcp}^e(x_{n,k}^{II})$ directly, when we treat $\xi = \sum_{\forall m \neq n} h_{m,D}^{II} \sum_{\forall k} \rho_k x_{m,k}^{II} + h_{n,D}^{II} \sum_{\forall \kappa \neq k} \rho_m x_{n,\kappa}^{II}$ as Gaussian noise. As a result, the interference cancellation algorithm discussed above may be reused. As BPSK modulation is assumed, the equality $\mathcal{L}_{dcp}^e(c_{n,k}^{II}) = \mathcal{L}_{dcp}^e(x_{n,k}^{II})$ holds.

The decomposition operation of CNC may be viewed as the decoding of a check node having $(K+1)$ edges. In general, let $\mathcal{L}_i^e$ denote the *extrinsic* LLR along the ith edge of a check node. Then, given the *a priori* LLRs $\mathcal{L}_j^a, j \neq i \in [1, K+1]$ of the rest of the edges of the check node, the typical sum-product algorithm for a check node may be formulated as [8] $\mathcal{L}_i^e = \sum_{\forall j \neq i} \boxplus \mathcal{L}_j^a$, where the operation $\boxplus$ was introduced for notation convenience in [9]. Hence the *extrinsic* LLR forwarded to the DEC of source k at relay n and the *extrinsic* LLR fed back to the DET are given by

$$\mathcal{L}_{dcp}^e(c_{n,k}^{II}) = \sum_{\forall \kappa \neq k} \boxplus \mathcal{L}_{dcp}^a(c_{n,\kappa}^{II}) \boxplus \mathcal{L}_{det}^e(c_n^{II}) \tag{10}$$

$$\mathcal{L}_{det}^a(c_n^{II}) = \sum_{\forall k} \boxplus \mathcal{L}_{dcp}^a(c_{n,k}^{II}), \tag{11}$$

where we have $\mathcal{L}_{det}^e(c_n^{II}) = \mathcal{L}_{det}^e(x_n^{II})$, when BPSK modulation is assumed and hence $\mathcal{L}_{det}^a(x_n^{II}) = \mathcal{L}_{det}^a(c_n^{II})$ is used as the *a priori* LLR for the DET block of Fig 3.

The decomposition operation of our packet-wise VNC scheme involves message passing between the check and the variable nodes, as seen in the right subplot of Fig 2. During a particular self-iteration, when processing the ith bit sequence, let $\mu^a_{v\to c}$ and $\mu^a_{c\to v}$ denote the messages passed from the variable node to the check node and from the check node to the variable node, respectively, where the superscripts of $a = 0$ and $a = 1$ indicate the message passed along the current bit sequence and the adjacent bit sequence, respectively, we have:

$$\mu^0_{v\to c}(i) = \mu^a[c^{II}_n(i)] + \mu^1_{c\to v}(i+1) \qquad (12)$$

$$\mu^1_{v\to c}(i) = \mu^a[c^{II}_n(i-1)] + \mu^0_{c\to v}(i-1) \qquad (13)$$

$$\mu^e[c^{II}_{n,T}(i)] = \mu^0_{v\to c}(i) \boxplus \mu^1_{v\to c}(i) \qquad (14)$$

$$\mu^0_{c\to v}(i) = \mu^a[c^{II}_{n,T}(i)] \boxplus \mu^1_{v\to c}(i) \qquad (15)$$

$$\mu^1_{c\to v}(i) = \mu^a[c^{II}_{n,T}(i+1)] \boxplus \mu^0_{v\to c}(i+1) \qquad (16)$$

$$\mu^e[c^{II}_n(i)] = \mu^0_{c\to v}(i) + \mu^1_{c\to v}(i), \qquad (17)$$

where $\mu^e[c^{II}_{n,T}(1)] = \mu^0_{v\to c}(1)$, $\mu^a[c^{II}_{n,T}(1)] = \mu^0_{c\to v}(1)$ and $\mu^e[c^{II}_n(KN_c)] = \mu^0_{c\to v}(KN_c)$, $\mu^a[c^{II}_n(KN_c)] = \mu^0_{v\to c}(KN_c)$. In the above algorithm, the *a priori* LLRs $\mu^a(c^{II}_{n,T})$ are the consequences of the P/S conversion of $\mathcal{L}^a_{dcp}(c^{II}_{n,k}), \forall k$ and the *extrinsic* LLRs $\mu^e(c^{II}_{n,T})$ are further subject to S/P conversion in order to forward each $\mathcal{L}^e_{dcp}(c^{II}_{n,k})$ value to the DEC block. On the other hand, the *a priori* LLRs $\mu^a(c^{II}_n)$ are generated from the *extrinsic* LLRs $\mathcal{L}^e_{det}(c^{II}_n) = \mathcal{L}^a_{det}(x^{II}_n)$ of the DET block representing the BPSK demodulator, which are interpolated as zeros, wherever the intermediate outputs of the VNC scheme were blocked. Furthermore, the *extrinsic* LLRs $\mu^e(c^{II}_n)$ are fed back to the DET block subject to enabling by the set Ω in order to yield $\mathcal{L}^a_{det}(c^{II}_n)$ and hence we have $\mathcal{L}^a_{det}(x^{II}_n) = \mathcal{L}^a_{det}(c^{II}_n)$ for BPSK modulation.

4) Decoding and Combination: After obtaining the *a priori* LLRs $\mathcal{L}^{a,c}_{dec}(c^{II}_{n,k})$ of the coded bits, which are the deinterleaved versions of $\mathcal{L}^e_{dcp}(c^{II}_{n,k})$, a typical soft decoder algorithm is employed in the DEC block in order to generate the *extrinsic* LLRs $\mathcal{L}^{e,u}_{dec}(b^{II}_{n,k})$ of the uncoded information bits by taking into account the *a priori* LLRs $\mathcal{L}^{a,u}_{dec}(b^{II}_{n,k})$ of the uncoded information bits gleaned from the COM block. Then, the different versions of $\mathcal{L}^{e,u}_{dec}(b^{II}_{n,k})$ received from the N relays are combined to yield an improved estimate of $\mathcal{L}^{a,u}_{dec}(b^{II}_{n,k})$ by taking into account the soft LLRs obtained from the first transmission phase. In the feedback direction, the *a priori* LLRs $\mathcal{L}^{a,c}_{dec}(c^{II}_{n,k})$ of the coded bits and the improved *a priori* LLRs $\mathcal{L}^{a,u}_{dec}(b^{II}_{n,k})$ of the uncoded information bits gleaned from the output of the COM block are used to yield the *extrinsic* LLRs $\mathcal{L}^{e,c}_{dec}(c^{II}_{n,k})$ of the coded bits based on the corresponding soft decoding algorithm. After the last receiver iteration, the ultimate decision is made at the output of the COM block.

III. EXIT Chart Analysis

A. Configurations and Assumptions

We now investigate the iterative decoding convergence behaviour of our receiver during the second phase in conjunction with different multiple source processing techniques employing EXtrinsic Information Transfer (EXIT) charts [10], where the concatenation of the COM and DEC and the concatenation of the DCP and DET act as our outer and inner code.

- We consider a $K = N = 4$ MSC scenario, where all channels are AWGN channels. We assume perfect source-to-relay transmissions and Remark 2 is obeyed. An SNR per bit of $\gamma_0^I = 0dB$ and $\gamma_0^{II} = 6dB$ is assumed for the first phase source to destination transmission and second phase transmission is assumed, respectively.
- In the second phase, we maintain the same normalised throughput per relay node given by $\eta = 1$ for all methods considered. For the CDM and CNC method having $\eta_{cdm} = \eta_{cnc} = Kr$, we let $r = 1/K$. For our VNC scheme, no channel codes are employed and all intermediate outputs of the VNC were enabled. Further, in order to characterize the coding gain offered by the CNC and VNC schemes, we employ a repetition code of rate $r = 1/K$ as our channel coding scheme.

B. Convergence Analysis

1) Outer Code EXIT Curve: Since the COM block may be viewed as being equivalent to a repetition code of rate $r = 1/N$, when the CDM and CNC techniques are employed, the combined outer code is equivalent to a repetition code of rate $r = 1/NK$. When our VNC is considered, no DEC is employed, hence the combined outer code is equivalent to a repetition code of rate $r = 1/N$. Hence, the outer code may be interpreted as a variable node having $(d_v + 1)$ edges, where $d_v = 1/r$ and the additional edge is associated with the first phase direct source-to-destination transmission. As a result, the EXIT curve of the outer code may be formulated as [11]:

$$I^e_o(I^a_o, d_v, \gamma^I) = J\left(\sqrt{(d_v-1)[J^{-1}(I^a_o)]^2 + \sigma^2_{ch}}\right),$$

where I^a_o and I^e_o denote the *a priori* and *extrinsic* mutual information of the outer code, respectively. Further, $\sigma_{ch} = 2/\sigma^I_D$ is the variance of the uncoded source-to-destination channel LLR values, where we have $\sigma^I_D = \sqrt{1/2 \cdot 10^{\gamma^I_0/10}}$ [11].

2) Inner Code EXIT Curve for the CDM and CNC Methods: When CDM is considered, the EXIT curve of the composite inner code may only be obtained by simulation, in order to determine the relationship of $I^e_{cdm\&det}(I^a_{cdm\&det}, \gamma^{II}_0)$.

By contrast, the CNC method may be viewed as a $d_c = K$-degree check node having a total of $(d_c + 1)$ edges, where the additional edge is connected to the DET block. The EXIT curve of a d_c-degree check node may be accurately approximated with the aid of the EXIT curve of the $d_v = d_c$-degree variable node in an AWGN scenario by exploiting their duality [11]. Hence, when considering the soft information passing from the CNC to the DET block, we have [11]:

$$I^a_{det}(I^a_{cnc}, d_c + 1) \approx 1 - J\left(\sqrt{d_c[J^{-1}(1 - I^a_{cnc})]^2}\right),$$

where I^a_{det} and I^a_{cnc} denotes the *a priori* mutual information of DET and CNC. By contrast, the EXIT curve relation for the soft information flow from the CNC to the outer code is [11]:

$$I^e_{cnc}(I^a_{cnc}, I^e_{det}, d_c)$$
$$\approx 1 - J\left(\sqrt{(d_c-1)[J^{-1}(1 - I^a_{cnc})]^2 + [J^{-1}(1 - I^e_{det})]^2}\right),$$

where I^e_{det} and I^e_{cnc} denotes the *extrinsic* mutual information of DET and CNC. Furthermore, the EXIT curve of the DET block requires simulations in order to determine the relationship of $I^e_{det}(I^a_{det}, \gamma_0^{II})$. Hence the composite inner code EXIT curve of CNC is generated as:

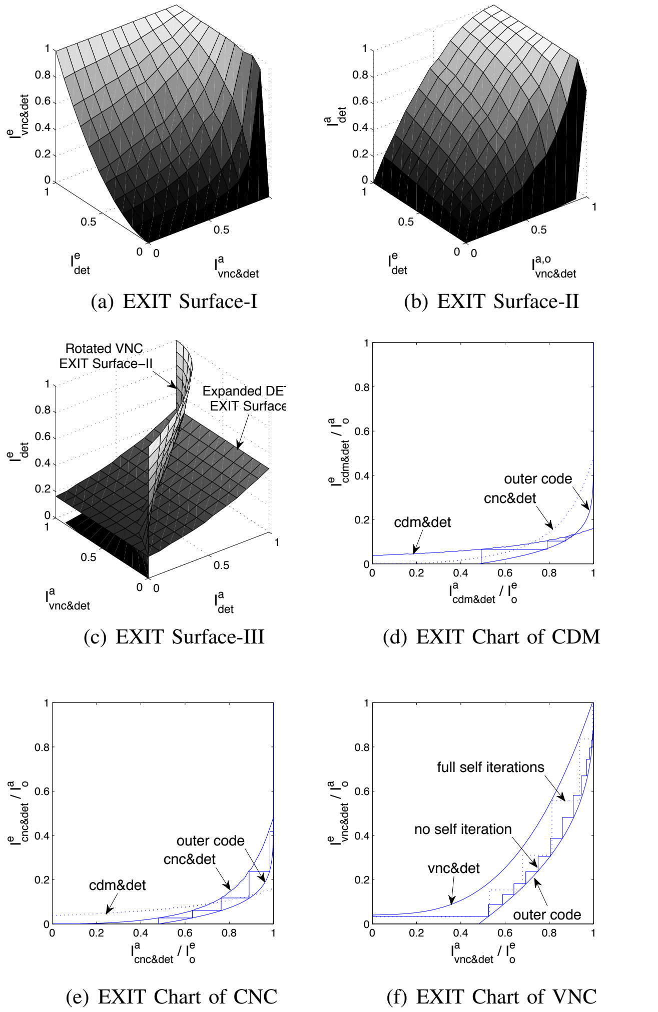
$$\begin{aligned} & I^e_{cnc\&det}(I^a_{cnc\&det}, \gamma_0^{II}, d_c) \\ & = I^e_{cnc}[I^a_{cnc\&det}, I^e_{det}(I^a_{det}, \gamma_0^{II}), d_c] \\ & = I^e_{cnc}[I^a_{cnc\&det}, I^e_{det}(I^a_{det}(I^a_{cnc\&det}, d_c+1), \gamma_0^{II}), d_c]. \end{aligned}$$

3) Inner Code EXIT Curve for the VNC Method: The EXIT curve of VNC depends on the number of self-iterations employed, which affects the resultant combined EXIT curve of the VNC and DET blocks. Hence, we plot the EXIT curve of VNC for a sufficiently high number of self-iterations as the best-case benchmarker. The block of VNC has two *a priori* inputs and two *extrinsic* outputs, the *extrinsic* outputs become not only a function of the two *a priori* inputs but also of the soft information generated during the self iterations. Therefore, we employ three-dimensional EXIT charts to visualise the iterative decoding convergence behaviour of our VNC.

We first draw the EXIT surface-I of the *extrinsic* output $I^e_{vnc\&det}$ forwarded to the outer code as a function of two *a priori* inputs, namely of $I^a_{vnc\&det}$ provided by the outer code and I^e_{det} provided by the DET block, as seen in Fig 4(a). We then draw the EXIT surface-II of the *extrinsic* output I^a_{det} provided for the DET block as a function of two *a priori* inputs, namely of $I^a_{vnc\&det}$ and I^e_{det} recorded from our simulations, as seen in Fig 4(b). We also draw the EXIT curve representing the *extrinsic* output I^e_{det} as a function of its *a priori* input I^a_{det} and then expand it into EXIT surface-III by incorporating another dimension, namely $I^a_{vnc\&det}$, which has no affect on the EXIT relationship between I^e_{det} and I^a_{det}, as demonstrated by the flat surface of Fig 4(c).

When having three EXIT surfaces, we first rotate EXIT surface-II by portraying I^e_{det} as a function of $I^a_{vnc\&det}$ and I^a_{det}. The rotated EXIT surface-II is then plotted together with EXIT surface-III that has the same axes as seen in Fig 4(c). Thus, the curve of intersection between these two EXIT surfaces emerges. By recording the curve of intersection described by the three-triple $\{I^a_{det}, I^a_{vnc\&det}, I^e_{det}\}$ and mapping them onto EXIT surface-I, we get $I^e_{vnc\&det}$ with the two *a priori* inputs of I^e_{det} and $I^a_{vnc\&det}$ provided by the curve of intersection. In this way, we obtain the relationship between $I^e_{vnc\&det}$ and $I^a_{vnc\&det}$ as the inner code's EXIT curve.

4) Convergence Analysis: Fig 4(d) and Fig 4(e) show the EXIT charts of the CDM technique and the CNC technique, respectively. Observe in Fig 4(d) that the CDM method is unable to achieve iterative convergence while the CNC method is capable of achieving convergence as seen in Fig 4(e). More explicitly, we can see that the EXIT curve of the CNC has a lower *extrinsic* mutual information in the low *a priori* mutual information region than that of the CDM, while it exhibits a higher *extrinsic* mutual information in the high *a priori* mutual information region. This follows the typical behaviour of check nodes. Note that the EXIT curve of the CNC method emerges from the origin, hence it fails to converge, if no first phase direct source-to-destination transmission is available.



(a) EXIT Surface-I

(b) EXIT Surface-II

(c) EXIT Surface-III

(d) EXIT Chart of CDM

(e) EXIT Chart of CNC

(f) EXIT Chart of VNC

Fig. 4. The EXIT curves for various multiple source computation methods.

Fig 4(f) shows the EXIT curve of the r_1-VNC scheme employing a sufficiently high number of self-iterations. The VNC scheme is capable of achieving decoding convergence, since it reaches the EXIT chart's top right corner at the [1,1] point. In addition, we also recorded the Monte-Carlo simulation based decoder trajectories, when employing no self-iterations in VNC decoding and when employing a sufficiently high number of self-iterations during VNC decoding, which are represented by the stair-case-shaped solid trajectory and dotted trajectory, respectively. It can be seen in Fig 4(f) that both self-iteration configurations achieve convergence, where the arrangement using no self-iterations evolves the mutual information in its own way, embedded in the iteration loop constituted by the outer and inner code. Another important property is that the VNC EXIT curve is shifted upwards from the origin, implying that the VNC method is capable of operating even without the first phase direct transmission.

IV. Performance Evaluation

Before presenting our simulation results, the following configurations and assumptions are made:

- We consider a $K = N = 4$ MSC scenario communicating over quasi-static flat Rayleigh fading channels. Remark 2 is obeyed and we assume $E_b/N_0 = \gamma_0^I = \gamma_0^{II}$.

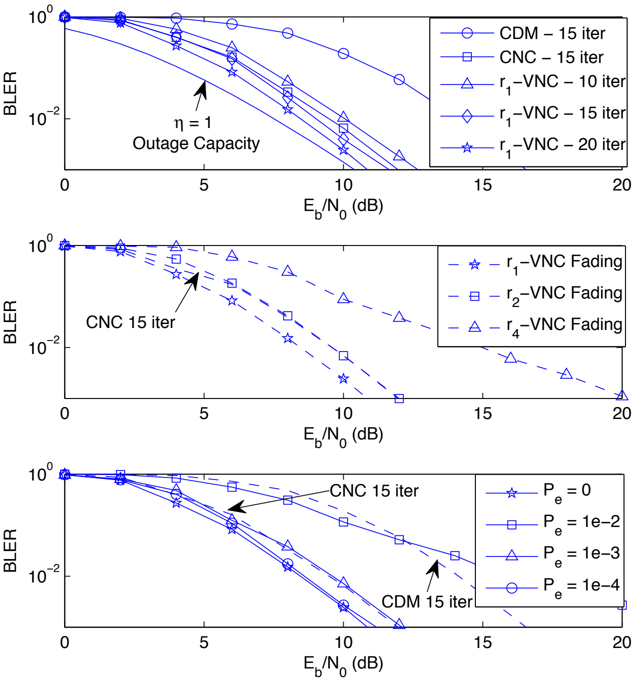


Fig. 5. The simulation results of the effect of iterations (top), variable rates (middle) and decoding errors (bottom).

- During our simulations, the BLER versus E_b/N_0 characteristics were recorded and each source had a packet length of $N_b = 1024$. We also employed the arrangement using no self-iterations for the decoding of VNC.
- Apart from the middle plot of Fig 5, the second assumption stipulated in Section III-A is also exploited.

1) Effect of Iterations: The top of Fig 5 show the effects of iterations at the destination receiver for various multiple source processing methods employed at the relays. For the CDM and CNC methods, the attainable performance was recorded after 15 iterations, beyond which no further performance improvements may be achieved. The CDM method represented by the circle legends exhibits the worst performance, while the element-wise CNC technique characterized by the square legends is capable of attaining a coding gain of about 4dB at BLER of 0.001. Furthermore, the performance of r_1-VNC relying on 10 iterations and denoted by the triangle legend is inferior to that of the CNC method. However, when 15 iterations are employed denoted by the rhombus legend, the VNC becomes superior in comparison to its CNC counterpart and attains a further performance gain of about 1dB when using 20 iterations as indicated by the star legend at BLER of 0.001. Importantly, we also plot the outage capacity curve at a rate of $\eta = 1$ for the quasi-static Rayleigh fading channel having a transmit diversity order of $(N + 1) = 5$, where each of the $N = 4$ diversity branches has a normalised power of $P_n = 1/8$, while one of the additional diversity branches has a normalised power of $P^{II} = 1/2$. It can be seen that the r_1-VNC using 20 iterations attains a BLER performance that is less than 1dB from the outage capacity.

2) Effect of Variable Rate: The middle of Fig 5 shows the BLER versus E_b/N_0 performance of our VNC method for different number of intermediate outputs, where the destination receiver uses 20 iterations. More explicitly, every second and every fourth intermediate outputs are enabled in the r_2-VNC and r_4-VNC schemes, respectively. Unsurprisingly, increasing the normalised throughput per relay node by enabling less intermediate outputs leads to a BLER performance degradation. Remarkably, the r_2-VNC method exhibits a similar BLER performance, when compared to the CNC method at a doubled normalised throughput per relay node.

3) Effect of Relay Decoding Error: The bottom of Fig 5 shows the BLER versus E_b/N_0 performance for our r_1-VNC method in the presence of first phase decoding errors according to the Bit Error Probability (BEP) of P_e during the source-to-relay transmissions, where the destination receiver uses 20 iterations. We observed that the ultimate BLER performance substantially degrades at $P_e = 0.01$, as denoted by the square legend. At $P_e = 0.001$, the BLER performance degrades only gently and it performs similarly to the CNC method. At $P_e = 0.0001$, only a marginal performance degradation is observed. These investigations suggest that if a powerful channel code is employed for the first phase transmission, the second phase transmissions may be safely activated.

V. Conclusion

In this paper, multiple source processing techniques were investigated. We proposed a soft decoding method for CNC and a novel packet-wise VNC. Furthermore, the EXIT chart analysis of three different multiple source processing methods was carried out and their convergence behaviour was investigated. Our simulation results suggest that our proposed VNC provides a high flexibility and a near-optimal performance.

References

[1] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *submitted to IEEE Transactions on Information Theory*. [Online]. Available: http://arxiv.org/abs/0908.2119

[2] O. Shalvi, "Multiple source cooperation diversity," *IEEE Communications Letters*, vol. 8, pp. 712–714, Dec. 2004.

[3] A. Ribeiro, R. Q. Wang, and G. B. Giannakis, "Multi-source cooperation with full-diversity spectral-efficiency and controllable-complexity," *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 415–425, Feb. 2007.

[4] R. Zhang and L. Hanzo, "Coding schemes for energy efficient multi-source cooperation aided uplink transmission," *IEEE Signal Processing Letters*, vol. 16, pp. 438–441, May 2009.

[5] ——, "Interleaved random space-time coding for multisource cooperations," *IEEE Transactions on Vehicular Technology*, vol. 58, pp. 2120–2125, May 2009.

[6] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 4, pp. 1204–1216, Jul 2000.

[7] M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asenstorfer, "Iterative multiuser detection for CDMA with FEC: Near-single-user performance," *IEEE Transactions on Communications*, vol. 46, pp. 1693–1699, Dec. 1998.

[8] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, Feb. 2001.

[9] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Communications Magazine*, vol. 42, pp. 429–445, Mar. 1996.

[10] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727–1737, Oct. 2001.

[11] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Transactions on Communications*, vol. 52, pp. 670–678, Apr 2004.