Signal Processing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Approximate decoding for network coded inter-dependent data

Q1 Minhae Kwon [a], Hyunggon Park [a], Nikolaos Thomos [b], Pascal Frossard [c]

Q2 [a] Department of Electronics Engineering, Ewha Womans University, Seoul, South Korea
[b] Computer Science and Electronic Engineering Department, University of Essex, Colchester, United Kingdom
Q3 [c] Signal Processing Laboratory (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 17 December 2014
Received in revised form
12 July 2015
Accepted 14 September 2015

Keywords:
Network coding
Approximate decoding
Inter-dependence source
Ad hoc networks
Loss-tolerant data

ABSTRACT

In this paper, we consider decoding of loss tolerant data encoded by network coding and transmitted over error-prone networks. Intermediate network nodes typically perform the random linear network coding in a Galois field and a Gaussian elimination is used for decoding process in the terminal nodes. In such settings, conventional decoding approaches can unfortunately not reconstruct any encoded data unless they receive at least as many coded packets as the original number of packets. In this paper, we rather propose to exploit the incomplete data at a receiver without major modifications to the conventional decoding architecture. We study the problem of approximate decoding for inter-dependent sources where the difference between source vectors is characterized by a unimodal distribution. We propose a mode-based algorithm for approximate decoding, where the *mode* of the source data distribution is used to reconstruct source data. We further improve the mode-based approximate decoding algorithm by using additional short information that is referred to as position similarity information (PSI). We analytically study the impact of PSI size on the approximate decoding performance and show that the optimal size of PSI can be determined based on performance requirements of applications. The proposed approach has been tested in an illustrative example of data collection in sensor networks. The simulation results confirm the benefits of approximate decoding for inter-dependent sources and further show that 93.3% of decoding errors are eliminated when the approximate decoding uses appropriate PSI.

1. Introduction

The hardware development of sensors and communication chipsets has enabled easy deployment of sensor networks and it has led to an excessive network traffic and demands to increase network capacity. Network coding [1] has been proposed in order to increase the throughput of networks; it can reach the max-flow capacity between the source and each destination node [2–4]. In this case, unlike simple data forwarding in conventional networks, intermediate network nodes combine the received packets with basic coding operations. Network coding can lead to efficient resources usage (e.g., bandwidth and power), reduced computations, and improved robustness against network dynamics [5] by exploiting the diversity in networks. A variety of applications have been developed by taking advantages of network coding (e.g., content distribution, storages, and P2P systems [6–10]). Random linear network coding (RLNC) [11] is the most popular network coding algorithm, as it permits distributed deployment in dynamic error-prone networks [12].

E-mail addresses: minhae.kwon@ewhain.net (M. Kwon), hyunggon.park@ewha.ac.kr (H. Park), nthomos@essex.ac.uk (N. Thomos), pascal.frossard@epfl.ch (P. Frossard).

http://dx.doi.org/10.1016/j.sigpro.2015.09.010
0165-1684/© 2015 Elsevier B.V. All rights reserved.

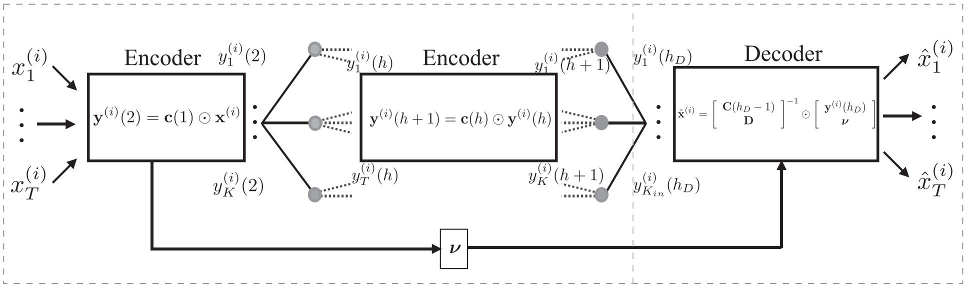

Fig. 1. Overall description of system setup. At the h-th coding stage, the incoming T innovative packets $y_m^{(i)}(h)$ $(1 \le m \le T)$ are combined based on RLNC and K outgoing packets are generated. If T innovative packets are not available for the decoder at the moment of decoding (i.e., $K_{in} < T$), the proposed approximate decoding is deployed with side information ν, which is delivered from the encoder. These are discussed in Sections 2.2 and 2.3, respectively.

While several advantages can be obtained by deploying network coding techniques for information delivery, it has a significant drawback in practice, which is also known as *all-or-nothing problem*, i.e., a terminal node cannot recover any information from the received data unless it receives at least the same number of innovative coded packets[1] as the number of source packets. In other words, under the conventional decoding process (e.g., Gaussian elimination), the received packets have to form a full-rank system for decoding. However, perfect decoding might not always be necessary and approximate reconstruction may be sufficient for several services that can accept imperfect reconstruction.

In order to solve this problem, we propose an approach to approximately recover inter-dependent sources from a set of network coded data that does not form a full rank system at receiver. With this same objective, a low complexity approximate decoding algorithm has been presented in [14], where the receiver simply matches the most similar data between neighbor sources and thus reaches only limited approximate decoding performance. In this paper, we present an improved approximate decoding algorithm that exploits the source characteristics, i.e., the distribution of differences between neighbor source vectors, thereby explicitly considering more general types of source data. We propose to use the *mode* of the distribution (i.e., the value that appears most often in a distribution) in the source characteristics to build an approximate decoding algorithm. The mode of the distribution is referred to as *similarity information* (SI). We show that it is sufficient side information to maximize performance of the proposed approximate decoding. As a result, the mode-based approximate decoding can significantly reduce the amount of side information needed for decoding. The decoding performance can be further improved by considering the positions where errors may occur, which are explicitly captured by the *position similarity information* (PSI) at the expense of additional side information. We investigate the tradeoff between the PSI size (i.e., the amount of side information or communication overheads) and the corresponding decoding performance and show that there is an optimal amount of additional information for approximate decoding. Finally, the proposed approach is deployed in an illustrative example of sensor networks and the simulation results confirm our theoretical performance study.

The main contributions of the paper can be summarized as follows:

- we propose a generalized framework of approximate decoding that covers large range of source types,
- we develop an algorithm that enables the approximate decoding solution to be deployed for any linearly inter-dependent sources,
- we develop decoding algorithms that can exploit both SI and PSI, leading to significantly improved decoding performance,
- we analytically study the tradeoff between communication overhead (incurred by deploying SI and PSI) and decoding performance gains, and
- we have extensive set of experiment results that confirm the theoretical analysis.

The rest of the paper is organized as follows. The general network coding framework is presented in Section 2. The mode-based decoding approximate decoding algorithm that considers inter-dependent source distributions is proposed and discussed in Section 3. In Section 4, we show that the decoding performance can be improved by incorporating PSI into the mode-based approach. In Section 5, we evaluate and compare the performance of the mode-based approach against conventional decoding methods in an illustrative sensor network scenario. Related works are discussed in Section 6 and conclusion is drawn in Section 7.

2. System setup

We consider data transmission over error-prone networks that consist of source nodes, intermediate nodes and receivers. The source data is delivered to the receivers through intermediate nodes that are able to perform network coding, similar to the frameworks in [15–17]. The overview of the proposed system is shown in Fig. 1 and the details will be discussed next.

[1] A packet is *innovative* for a node if its coding vector is not in the span of the coding vectors of the packets already available at the node [13].

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

2.1. Source description

Consider T source vectors $\mathbf{s}_t$ $(1 \le t \le T)$ and let $\mathbf{s}_t$ be the t-th measured symbol vector that consists of L symbols denoted by $s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(L)}$. These symbols are discrete, i.e., $s_t^{(i)} \in \mathbb{R}$ for $1 \le i \le L$, and inter-dependent. $\mathbb{R}$ represents the field of real numbers. Since the network coding operations are performed in the Galois field (GF), each $s_t^{(i)}$ has to be discretized and mapped into an element in GF. This operation is represented by a function $\mathcal{Q}$ that transforms the source data to GF with size 2^M, denoted by $GF(2^M)$, where M is a positive integer. The function $\mathcal{Q}: \mathbb{R} \to GF(2^M)$ is defined as

$$\mathcal{Q}\left(s_t^{(i)}\right) = x_t^{(i)} \in \mathcal{X} \tag{1}$$

where $x_t^{(i)}$ is the GF representation of $s_t^{(i)}$ in its alphabet set $\mathcal{X}$. Note that the GF size is determined by M that can be appropriately chosen by considering the sizes of the source alphabet, maximum bandwidth constraints, etc. Then, the measured symbol vector $\mathbf{s}_t = [s_t^{(1)}, \ldots, s_t^{(L)}]^T$ is mapped into $\mathbf{x}_t = [x_t^{(1)}, \ldots, x_t^{(L)}]^T$ by $\mathcal{Q}$, where the notation $(\cdot)^T$ represents the matrix transpose operator. In order to simplify the notations used in this paper, the field of operands is implicitly assumed to be the same as the field of operators. For example, if an operator is in a GF, all the associated operands are implicitly assumed to be elements in the GF. The operators $\oplus$ and $\otimes$ denote the addition and multiplication in GF, respectively, and the operator $\odot$ represents the multiplication between matrices in GF. In this paper, $\oplus$ in GF is performed by the logical bitwise exclusive-OR (XOR) operator. Conversely, if an operator is in $\mathbb{R}$ (e.g., $+$, $-$, $\times$), the operands are also in $\mathbb{R}$.

2.2. RLNC based encoding

In the network, an intermediate node deploys RLNC and generates coded packets. Let $\mathbf{y}^{(i)}(h) = [y_1^{(i)}(h), \ldots, y_m^{(i)}(h), \ldots, y_T^{(i)}(h)]^T$ be incoming packets at the h-th coding stage; $\mathbf{y}^{(i)}(1) = [x_1^{(i)}, \ldots, x_T^{(i)}]^T$ is the initial source data packet. At the h-th coding stage intermediate node, the first T innovative packets (e.g., $y_m^{(i)}(h)$ for $1 \le m \le T$) are combined using RLNC and generate K outgoing packets (e.g., $y_k^{(i)}(h+1)$ for $1 \le k \le K$) as

$$\mathbf{y}^{(i)}(h+1) = \mathbf{c}(h) \odot \mathbf{y}^{(i)}(h) \tag{2}$$

or equivalently,

$$\begin{bmatrix} y_1^{(i)}(h+1) \\ \vdots \\ y_k^{(i)}(h+1) \\ \vdots \\ y_K^{(i)}(h+1) \end{bmatrix} = \begin{bmatrix} c_{11}(h) & \cdots & c_{1m}(h) & \cdots & c_{1T}(h) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{k1}(h) & \cdots & c_{km}(h) & \cdots & c_{kT}(h) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{K1}(h) & \cdots & c_{Km}(h) & \cdots & c_{KT}(h) \end{bmatrix}$$
$$\odot \begin{bmatrix} y_1^{(i)}(h) \\ \vdots \\ y_m^{(i)}(h) \\ \vdots \\ y_T^{(i)}(h) \end{bmatrix} \triangleq \sum_{m=1}^{T} \oplus \{\mathbf{c}_m(h) \otimes y_m^{(i)}(h)\}.$$

The number of outgoing packets K is chosen larger than T (i.e., the number of symbols combined in a packet) and K may depend on the expected packet erasure rate; higher K is recommended for higher erasure rate and vice versa. In terms of element-wise operations, $y_k^{(i)}(h+1)$ is generated as

$$y_k^{(i)}(h+1) = \sum_{m=1}^{T} \oplus \left\{c_{km} \otimes y_m^{(i)}(h)\right\} = \{c_{k1}(h) \otimes y_1^{(i)}(h)\} \oplus \cdots$$
$$\oplus \{c_{km}(h) \otimes y_m^{(i)}(h)\} \oplus \cdots \oplus \left\{c_{kT}(h) \otimes y_T^{(i)}(h)\right\}.$$

Each outgoing packet $y_k^{(i)}(h+1)$ is a linear combination of the incoming packets $y_m^{(i)}(h)$ for $1 \le m \le T$ with coding coefficients $c_{km}(h)$. The number of combined symbols is denoted by T and $\mathbf{c}_m(h)$ is a coding coefficient vector that is defined as

$$\mathbf{c}_m(h) = \left[c_{1m}(h), \ldots, c_{km}(h), \ldots, c_{Km}(h)\right]^T.$$

In this paper, network coding is implemented with RLNC so that the coding coefficients are uniformly and randomly chosen from $GF(2^M)$, i.e., $c_{km}(h) \in GF(2^M)$.

We finally note that the coded packet at the h-th coding stage of the network in (2) can be expressed as

$$\mathbf{y}^{(i)}(h+1) = \mathbf{c}(h) \odot \mathbf{y}^{(i)}(h) = \mathbf{c}(h) \odot \mathbf{c}(h-1) \odot \cdots \odot \mathbf{c}(1)$$
$$\odot\, \mathbf{x}^{(i)} = \mathbf{C}(h) \odot \mathbf{x}^{(i)}$$

where $\mathbf{C}(h)$ is referred to as a global coding coefficient matrix [13], which is included in the header of the packet and delivered to the decoder to enable decoding and reconstruction. $\mathbf{c}(h)$ can be selected such that $\mathbf{C}(h)$ is invertible with high probability in RLNC [11].

In the next section, we focus on designing an approximate decoding approach, which can be deployed if a rank deficient system of equation is available at the receiver. Note that the system becomes rank deficient because of the packet loss and delay, but not because of the random selection of $\mathbf{c}(h)$ (as $\mathbf{C}(h)$ is assumed to be full-rank).

2.3. Data reconstruction based on approximate decoding

With the coding procedure described above, K coded packets are generated at intermediate network nodes using T innovative incoming packets ($K \ge T$) and traverse error-prone network toward the destination node h_D. When a decoder receives K_{in} innovative packets,[2] it attempts to recover the source data $\hat{x}^{(i)}$. In the decoding process, well-known Gaussian elimination in the considered GF [18] is employed for matrix inversion and if the $K_{in} \times T$ global coding coefficient matrix $\mathbf{C}(h_D - 1)$ is full-rank (i.e., $K_{in} = T$), then $\hat{\mathbf{x}}^{(i)} = [\hat{x}_1^{(i)}, \ldots, \hat{x}_T^{(i)}]^T$ is uniquely determined by

$$\hat{\mathbf{x}}^{(i)} = [\hat{x}_1^{(i)}, \ldots, \hat{x}_T^{(i)}]^T = \mathbf{C}(h_D - 1)^{-1} \odot \mathbf{y}^{(i)}(h_D) \tag{3}$$

because the inverse of a full-rank global coding coefficient matrix $\mathbf{C}(h_D - 1)^{-1}$ is unique.

If the number of received packets is insufficient to determine $\mathbf{C}(h_D - 1)^{-1}$ uniquely, i.e, in the presence of a singular matrix $\mathbf{C}(h_D - 1)$ matrix (e.g., due to packet loss or

[2] Note that $K_{in} \le K$ and it depends on packet erasure of network condition.

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

Table 1
Summary of Notations.

Notation	Description	Notation	Description
h	Coding stage index	δ_t	Result vector of $\mathbf{x}_{t+1} - \mathbf{x}_t$
h_D	Final stage (destination node) index	$\delta_t^{(i)}$	i-th element of δ_t
T	The number of innovative packets encoded together	Ψ_Δ	Distribution of inter-dependent source with mode Δ
K	The number of outgoing packets generated at an intermediate node ($K \geq T$)	Δ	Similarity Information (SI)
K_{in}	The number of innovative incoming packets at a node ($K_{in} \leq K$)	$\mathbf{\Delta}_t$	Result vector of $\mathbf{x}_t \oplus \mathbf{x}_{t+1}$
$\mathbf{s}_t$	t-th measured symbol vector in $\mathbb{R}^L$	Δ_n	n-th candidate for Δ in GF
$s_t^{(i)}$	i-th element of $\mathbf{s}_t$ in $\mathbb{R}$	Δ_R	Random variable for Δ_n
$\mathbf{x}_t$	t-th source vector	$GF(2^M)$	Galois field with size of 2^M
$x_t^{(i)}$	i-th element of $\mathbf{x}_t$ in GF	L	Length of a source vector
$\mathbf{y}^{(i)}(h)$	i-th incoming encoded data vector at the h-th coding stage	$\mathbf{c}(h)$	Coding coefficient matrix (with size $K \times T$) at the h-th coding stage
$y_m^{(i)}(h)$	m-th element of $\mathbf{y}^{(i)}(h)$ in GF	$c_{km}(h)$	k-th element of $\mathbf{c}_m(h)$
$\mathbf{c}_m(h)$	m-th coding coefficient column vector of $\mathbf{c}(h)$	$\mathbf{C}(h)$	Global coding coefficient matrix

delays) with $K_{in} < T$, the receiver is not able to find the inverse of the coding coefficient matrix and there are multiple solutions to the system in (3). With additional constraints, a good but not-necessarily optimal solution can still be determined. In order to add constraints and find a unique solution, the idea of an approximate decoding approach is introduced in [14] that considers a simple source model and permits approximate reconstruction of sources from an insufficient number of received packets. The approximate decoding method adds simple constraints in order to form a full rank coding coefficient matrix, thereby enabling the decoding process from the system in (3) with external information. In particular, the inter-dependence between the input data is exploited in order to build additional constraints $\mathbf{D}$ and $\boldsymbol{\nu}$ which render the system in (4) solvable:

$$\hat{\mathbf{x}}^{(i)} = \begin{bmatrix} \mathbf{C}(h_D - 1) \\ \mathbf{D} \end{bmatrix}^{-1} \odot \begin{bmatrix} \mathbf{y}^{(i)}(h_D) \\ \boldsymbol{\nu} \end{bmatrix}. \quad (4)$$

The coefficients in $\mathbf{D}$ and $\boldsymbol{\nu}$ are in GF and depend on the source model. In this paper, $\boldsymbol{\nu}$ is assumed to be delivered with high reliability. For example, separate channels such as control channels can be used [19–22] or high level of error protection techniques can be deployed for $\boldsymbol{\nu}$.

A simple implementation of the approximate decoding method is presented in [14]. In particular, the matrix $\mathbf{D}$ of size $(T - K_{in}) \times T$ is constructed such that each of its rows consists of zeros (i.e., additive identity of $GF(2^M)$) except for two elements with value "1"[3] that correspond to the positions of the $x_t^{(i)}$ and $x_{t'}^{(i)}$ such that $x_t^{(i)} = x_{t'}^{(i)}$ in the received packets. Then, the vector $\boldsymbol{\nu}$ is determined accordingly as a zero vector with size $(T - K_{in})$ (i.e., $\boldsymbol{\nu} = \mathbf{0}_{T-K_{in}}$). This means that $x_t^{(i)}$ and $x_{t'}^{(i)}$ shall have the same value in the approximate decoding algorithm proposed in [14]. Finally, an approximation $\hat{\mathbf{x}}^{(i)}$ of the original data is obtained as

$$\hat{\mathbf{x}}^{(i)} = \begin{bmatrix} \mathbf{C}(h_D - 1) \\ \mathbf{D} \end{bmatrix}^{-1} \odot \begin{bmatrix} \mathbf{y}^{(i)}(h_D) \\ \mathbf{0}_{(T-K_{in})} \end{bmatrix} \quad (5)$$

where $\mathbf{0}_{(T-K_{in})}$ is a vector with $(T - K_{in})$ zeros. This approach permits the receiver to approximately reconstruct the original symbols when the number of symbols is not sufficient for perfect decoding.

The key idea of the approximate decoding method is to incorporate additional equations (i.e., $\mathbf{D}$) based on source model, such that the matrix $[\mathbf{C}(h_D - 1)^T \ \mathbf{D}^T]^T$ in (5) becomes full rank. Without changes in conventional decoding algorithm (e.g., Gaussian elimination), approximate decoding algorithm does not need complex processes such as belief propagation [23] and it can be easily used when the received data does not form a full rank system. While the approximate decoding algorithm in [14] shows a new paradigm of network coded data reconstruction, the additional equations in (5) are very simple and cannot fully capture complex inter-dependent source models. Therefore, the algorithm in [14] can be used only in limited source model. In order to overcome the limitation, we propose an improved algorithm for approximate decoding that explicitly considers the statistical characteristics of the sources.

For reader's convenience, we summarize notations and abbreviations frequently used in this paper in Tables 2 and 3, respectively.

3. Mode-based approximate decoding for inter-dependent sources

In this section, we develop a new approximate decoding algorithm that explicitly considers the source characteristics for data reconstruction with incomplete sets of received packets.

[3] Recall that 1 is also an additive inverse of 1 in $GF\left(2^M\right)$.

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

Table 2
Summary of abbreviations.

Notation	Description	Notation	Description
RLNC	Random Linear Network Coding	*GF*	Galois Field
$\mathbb{R}$	Field of Real Numbers	SI	Similarity Information, The mode of Ψ_Δ
PSI	Position Similarity Information	MSE	Mean Square Error
PMSE	Peak Mean Square Error		

3.1. Description of inter-dependent source distribution

We consider a set of inter-dependent discrete sources. The sources are characterized by the distribution (i.e., probability mass function) of the difference between values in source data vectors. Let $\boldsymbol{\delta}_t = [\delta_t^{(1)}, \ldots, \delta_t^{(L)}]^T$ be an L-element vector representing the difference between data of the t-th and $(t+1)$-th sources, i.e., $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$, respectively, and $\delta_t^{(i)}$ be the i-th element of $\boldsymbol{\delta}_t$. We assume that the elements in $\boldsymbol{\delta}_t$ follow a unimodal distribution Ψ_Δ where its mode is Δ. Here, the source data can be expressed as

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \boldsymbol{\delta}_t \tag{6}$$

where $\delta_t^{(i)} \sim \Psi_\Delta$, or equivalently,

$$x_{t+1}^{(i)} = x_t^{(i)} + \delta_t^{(i)} \quad \text{for all } 1 \le i \le L. \tag{7}$$

In order to describe the entire source vectors, a single or multiple Ψ_Δ can be deployed. The inter-dependent source model can be exploited in a wide range of applications such as brightness changes in pictures, temperature variances, and seismic signals at different sensors [24,25]. The unimodal distribution can further represent noise in source data. A unimodal distribution includes various stochastic models such as Gaussian, Laplacian, chi-square and Cauchy distribution. An illustrative example of the above source model is shown in Fig. 2.

3.2. Approximate decoding algorithm for inter-dependent sources

We now discuss how to design the constraints $\boldsymbol{\nu} = [\nu_1, \nu_2, \ldots, \nu_{T-K_{in}}]^T$ that complete the decoding system in (4) such that the performance of the approximate decoding is maximized given the inter-dependent source distribution Ψ_Δ. The constraints $\boldsymbol{\nu}$ can provide additional equations that render the underdetermined decoding system solvable. Since the inter-dependent sources are characterized by $\delta_t^{(i)} \sim \Psi_\Delta$, $\boldsymbol{\nu}$ needs to be designed such that it represents the source characteristics.

Let $\boldsymbol{\nu}^*$ be the optimal constraints $\boldsymbol{\nu}$ that lead to a perfect decoding in (4). It is therefore desired that $\boldsymbol{\nu} = \boldsymbol{\nu}^*$. If $\boldsymbol{\nu} \ne \boldsymbol{\nu}^*$, the more zeros in the vector of $\boldsymbol{\nu} - \boldsymbol{\nu}^*$, the better performance of approximate decoding algorithm [17]. An illustrative result is shown in Fig. 3, where source data vectors have inter-dependency parameterized with $\nu_n^* = 16$, i.e., $x_{t+1}^{(i)} = x_t^{(i)} + 16$ and RLNC is deployed in $GF(2^{10})$. The decoding is based on Gaussian elimination and the decoding performance is measured by Peak Mean Square Error (PMSE) [26] for various $\boldsymbol{\nu} = \nu_n \cdot \mathbf{1}_{(T-K_{in})}$ in (4). The average PMSE for 1000 independent experiments is shown in Fig. 3, where the source vector $\mathbf{x}_1$ is randomly and independently determined in every experiment. This result confirms that approximate decoding leads to the best performance only when $\nu_n^* = \nu$ among $1 \le \nu_n \le 32$. Contrary to the intuition that closeness of ν_n to ν_n^* would increase the decoding performance, there is no obvious relationship between $\nu_n^* - \nu_n$ and performance. Rather, the number of zeros in $\boldsymbol{\nu} - \boldsymbol{\nu}^*$ determines the performance of approximate decoding.[4] The analysis of the decoding performance behavior is presented in Appendix A.

The shape of decoding performance given ν_n^* (see Fig. 3 and Appendix A) is well captured by a cost model $\mathcal{C}(\nu_n; \nu_n^*)$, referred to as the Cauchy–Dirac delta function [27], defined as

$$\mathcal{C}(\nu_n; \nu_n^*) = \lim_{\gamma \to 0} (1 - f(\nu_n; \nu_n^*, \gamma)) = \begin{cases} 0 & \text{if } \nu_n = \nu_n^* \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

where the probability density function of the Cauchy distribution is given by

$$f(\nu_n; \nu_n^*, \gamma) = \frac{1}{\pi\gamma \left[1 + \left(\dfrac{\nu_n - \nu_n^*}{\gamma} \right)^2 \right]}. \tag{9}$$

Here, ν_n^* and γ are determined by the sources as ν_n^* denotes the parameter that specifies the position of the peak of the distribution and γ is a scale parameter that specifies the half-width at half-maximum. Since $f(\nu_n; \nu_n^*, \gamma)$ is a probability measure, $f(\nu_n; \nu_n^*, \gamma) \in [0, 1]$. Therefore, the expected cost, which represents the decoding error, with any inter-dependent source distribution Ψ_Δ, is expressed as

$$E\{\mathcal{C}(\nu_n; \nu_n^*)\} = \sum_{\nu_n \in \boldsymbol{\nu}} \mathcal{C}(\nu_n; \nu_n^*) \Pr(V_n = \nu_n) \tag{10}$$

where V_n is a random variable for $\nu_n \in \boldsymbol{\nu}$ and is characterized by Ψ_Δ. In Property 1, we show that SI (i.e., Δ) leads to the minimum expected cost given the source distribution Ψ_Δ.

Property 1. *Given the inter-dependent source distribution, the mode of distribution (SI) minimizes expected cost of approximate decoding.*

Proof. The goal of the proof is to show that $(\nu_n^*)_{opt} = \Delta$ is the minimizer of the expected cost. The corresponding optimization problem is given by

$$(\nu_n^*)_{opt} = \arg\min_{\nu_n^* \in \boldsymbol{\nu}} E\{\mathcal{C}(\nu_n; \nu_n^*)\} \tag{11}$$

$$(\nu_n^*)_{opt} = \arg\min_{\nu_n^* \in \boldsymbol{\nu}} \sum_{\nu_n \in \boldsymbol{\nu}} \mathcal{C}(\nu_n; \nu_n^*) \Pr(V_n = \nu_n) \tag{12}$$

$$(\nu_n^*)_{opt} = \arg\min_{\nu_n^* \in \boldsymbol{\nu}} \sum_{\nu_n \in \boldsymbol{\nu}, \nu_n \ne \nu_n^*} \Pr(V_n = \nu_n) \tag{13}$$

$$(\nu_n^*)_{opt} = \arg\min_{\nu_n^* \in \boldsymbol{\nu}} [1 - \Pr(V_n = \nu_n^*)]$$

[4] Note that PMSE for $\nu_n = \nu_n^*$ is not zero in this experiment result. This is because $\boldsymbol{\nu} = \nu_n^* \cdot \mathbf{1}_{(T-K_{in})}$ is not enough for perfect decoding in *GF* where network coding operations are performed, while it can provide enough information in $\mathbb{R}$. This is studied in Property 3.

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

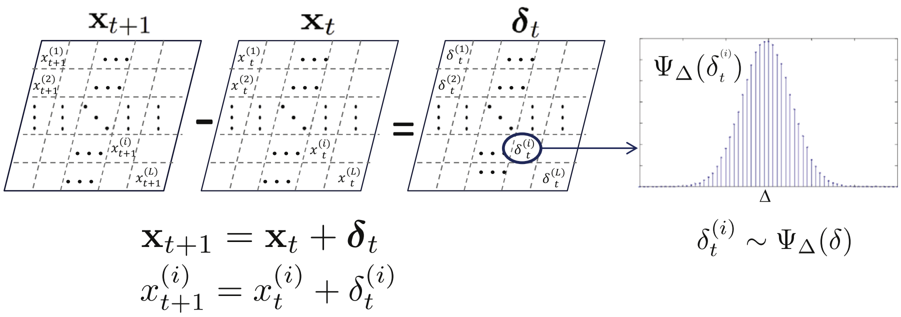

Fig. 2. An illustrative example of matrix-shaped inter-dependent sources and corresponding difference $\boldsymbol{\delta}_t$. $\delta_t^{(i)}$ is an element of $\boldsymbol{\delta}_t$ and $\delta_t^{(i)}$ follows a unimodal distribution Ψ_Δ where its mode is Δ.

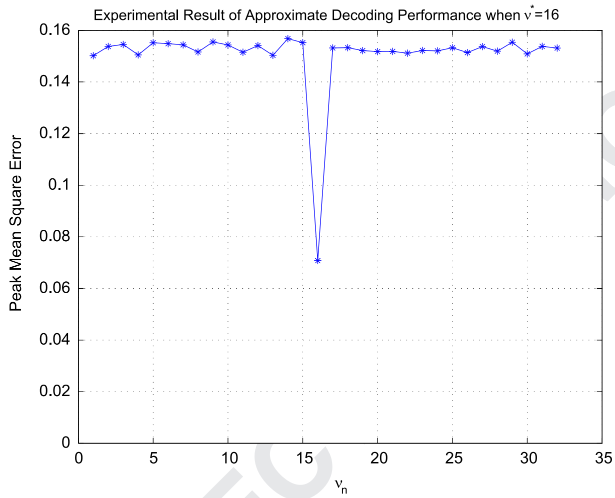

Fig. 3. An illustrative example for the average performance of approximate decoding with various ν_n when $\nu_n^* = 16$. The performance of approximate decoding is maximized only if $\nu_n = \nu_n^* = 16$. In this example, $GF(2^{10})$ is used and the sources are randomly and independently determined over 1000 times.

$$(\nu_n^*)_{opt} = \arg\max_{\nu_n^* \in \nu} \Pr(V_n = \nu_n^*) \tag{14}$$

$$(\nu_n^*)_{opt} = \Delta. \tag{15}$$

The equality between (12) and (13) comes from the definition of cost function given in (8), i.e., $\mathcal{C}(\nu_n; \nu_n^*) = 1$ if $\nu_n \neq \nu_n^*$ and $\mathcal{C}(\nu_n; \nu_n^*) = 0$ if $\nu_n = \nu_n^*$. Since $\Pr(V_n = \nu_n) \geq 0$, the minimization of $1 - \Pr(V_n = \nu_n^*)$ is equivalent to the maximization of $\Pr(V_n = \nu_n^*)$, leading to (14). Since V_n is characterized by the inter-dependent source distribution Ψ_Δ, $\max \Pr(V_n = \nu_n^*)$ is achieved only if $\nu_n^* = \Delta$. Therefore, $(\nu_n^*)_{opt} = \Delta$ minimizes the expected cost.□

The above property means that the decoder only needs to know Δ instead of the entire distribution Ψ_Δ in order to have effective decoding; the amount of the additional information that needs to be delivered is minimized.

This property is confirmed by the simulation results shown in Fig. 4. To generate an illustrative set of linearly inter-dependent source data (i.e., $x_{t+1}^{(i)} = x_t^{(i)} + \delta_t^{(i)}$), the source dependency $\delta_t^{(i)}$ is generated based on a Gaussian distribution $\mathcal{N}(16, \sigma^2)$ in Fig. 4(a), where the mean is 16 and the variance is σ^2. For a Gaussian distribution, the mode is the same as the mean, i.e., SI = 16($\Delta = 16$). For the sources in Fig. 4(b), $\delta_t^{(i)}$ is generated based on the Laplacian distribution $\mathcal{L}(16, \sigma^2/2)$ where the location parameter is 16 and the scale parameter is $\sigma^2/2$. For the Laplacian distribution, the mode is the same as the location parameter, i.e., SI = 16($\Delta = 16$). Hence, SI = $\arg\max_{\nu_n} \Pr(V_n = \nu_n) = 16$ in both cases. In these simulations, three source data vectors are combined together based on RLNC and 1 out of 3 packets is lost (i.e., 1/3 packet loss rate) so that approximate decoding strategies are deployed in $GF(2^8)$. The results are given in average PMSE computed from 1000 independent experiments. For the sake of comparison, we consider the following alternative three strategies [17]:

- *Strategy* 1 (*S*1): Select a single value Δ in Ψ_Δ (i.e., SI) as shown in Property 1, i.e., $\boldsymbol{\nu} = \Delta \cdot \mathbf{1}_{(T-K_{in})}$.
- *Strategy* 2 (*S*2): Sample the inter-dependent source distribution Ψ_Δ to select values of ν_n, i.e., $\nu_n \sim \Psi_\Delta$.
- *Strategy* 3 (*S*3): Sample a uniform distribution $\mathcal{U}(0, 2^8-1)$ to select values of ν_n, i.e., $\nu_n \sim \mathcal{U}(0, 2^8-1)$.

From the results, it is obvious that average PMSE is minimized by selecting SI (i.e., Strategy 1), which confirms Property 1.

In summary, under the knowledge of inter-dependent source distribution, SI for $\boldsymbol{\nu}$ leads to the minimized expected cost (i.e., $\boldsymbol{\nu} = \Delta \cdot \mathbf{1}_{(T-K_{in})}$). More importantly, it implies that the performance of the approximate decoding algorithm

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

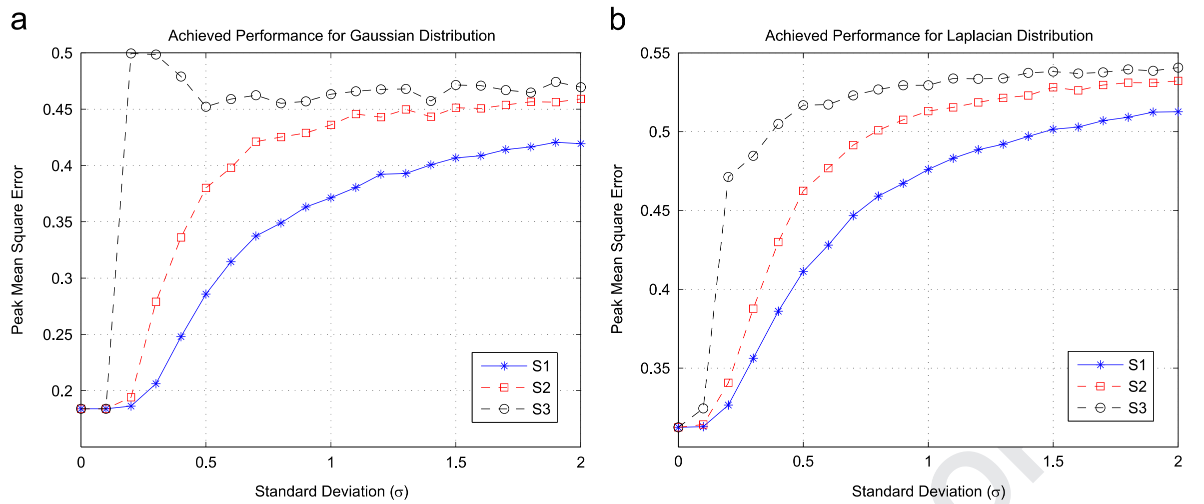

Fig. 4. Performance comparison of the three different strategies (S1, S2, S3). Sources are generated by (a) Gaussian and (b) Laplacian distributions with SI ($\Delta = 16$) and variance (σ^2). RLNC is deployed over $GF(2^8)$.

significantly depends on the number of ν_n that is exactly the same as ν_n^*. Hence, we will discuss how to maximize the number of zeros in $\nu_n - \nu_n^*$ in the next section.

4. Approximate decoding with PSI for linearly inter-dependent source data

In Section 3, it is discussed that the increased number of zeros in $\nu_n - \nu_n^*$ improves the performance of approximate decoding. In this section, our focus is on how to increase the number of zeros by deploying PSI constructed at the encoder as $\boldsymbol{\nu} = \boldsymbol{\Delta}_{PSI}$.

4.1. Δ-linearly inter-dependent source data and PSI

In Section 3, it is concluded that the approximate decoding with SI (i.e., the mode Δ) can minimize the average decoding error for unimodal inter-dependent source data. Therefore, the SI (i.e., the value of Δ) would be the only side information that needs to be transmitted to receivers if such decoding systems are implemented. From the decoder's perspective, the sources are seen as linearly and deterministically inter-dependent data in terms of SI. Here, we study the performance of the approximate decoding algorithm considering linearly and deterministically inter-dependent sources, which is also referred to as Δ-linearly inter-dependent source,[5] i.e.,

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta \cdot \mathbf{1}. \tag{16}$$

Note that if all the operations are performed in $\mathbb{R}$, the information Δ is sufficient for the perfect recovery of the original source data because $\mathbf{x}_{t+1} - \mathbf{x}_t = \Delta \cdot \mathbf{1}$. However, the corresponding operation in GF, $\mathbf{x}_{t+1} \oplus \mathbf{x}_t$, includes not only Δ but also the other outcomes. For example, when sources are described as $\mathbf{x}_t - \mathbf{x}_{t+1} = \Delta = 16$ in $\mathbb{R}$, $\mathbf{x}_t \oplus \mathbf{x}_{t+1}$ in GF includes not only 16 but also other values such as 48, 112, …, in $GF(2^8)$. Hence, we define $\boldsymbol{\Delta}_t$ as the vector of outcomes from $\mathbf{x}_{t+1} \oplus \mathbf{x}_t$, i.e., $\boldsymbol{\Delta}_t = \mathbf{x}_{t+1} \oplus \mathbf{x}_t$ where each element in $\boldsymbol{\Delta}_t$ is denoted by Δ_n. These elements are referred to as *candidates* of Δ. Since there may be different occurrences of Δ_n in $\boldsymbol{\Delta}_t$, we consider Δ_n as a random variable, is denoted by Δ_R.

As will be discussed later, $\Pr(\Delta_R = \Delta_n)$ is decreasing with respect to n. This means that the probability of Δ_1 is the highest among the candidates of Δ and $\Delta_1 = \Delta$ by definition. This is illustrated in Table 3 and the details are discussed in Property 3. Moreover, as discussed in Section 3.2, the accurate positions (or *indices*) of the candidates are the key information for improved performance of the proposed approximate decoding framework. Therefore, Strategy 1 (S1) in Section 3.2, which is shown to be the best implementation of approximate decoding for unimodal inter-dependent sources, can be further improved by considering the positions of the candidates $\boldsymbol{\Delta}_t$ included in PSI.

The PSI includes Δ_1 by default as it has the highest probability of occurrence in $\boldsymbol{\Delta}_t$. Then, it includes the position indices of Δ_n in $\boldsymbol{\Delta}_t$ for $n \geq 2$. The n-th row of PSI is filled with position indices of Δ_{n+1} in $\boldsymbol{\Delta}_t$. An illustrative example of PSI construction is shown in Fig. 5. If a decoder receives PSI, Strategy 1 (S1) that is based only on Δ is deployed as the baseline of approximate decoding, i.e., $\boldsymbol{\Delta}_{PSI} = \Delta_1 \cdot \mathbf{1}_{(T-K_{in})}$. Next, the elements with the position indices that are written in n-th row of PSI are replaced by Δ_{n+1} in $\boldsymbol{\Delta}_{PSI}$. Finally, $\boldsymbol{\nu}$ in (4) is replaced by $\boldsymbol{\Delta}_{PSI}$, leading to

$$\hat{\mathbf{x}}^{(i)} = \begin{bmatrix} \mathbf{C}(h_D - 1) \\ \mathbf{D} \end{bmatrix}^{-1} \odot \begin{bmatrix} \mathbf{y}^{(i)}(h_D) \\ \boldsymbol{\Delta}_{PSI} \end{bmatrix}. \tag{17}$$

In the next section, we study the properties of proposed approximate decoding with PSI that helps us to increase performance.

[5] In this paper, we assume that $\Delta \neq 0$ in GF, since $\Delta = 0$ simply means that two neighbor data vectors $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ are identical. Thus, if $\mathbf{x}_{t+1}$ is lost, it can be easily recovered simply by duplicating $\mathbf{x}_t$. For simplicity, we assume that the largest data is denoted by $\mathbf{x}_t$ which does not exceed the size of GF. Hence, $\mathbf{x}_t < 2^M \cdot \mathbf{1}$ for $GF(2^M)$.

4.2. *Properties of approximate decoding with psi for Δ-linearly inter-dependent source*

In this section, we focus on the investigation of the basic properties of the proposed approximate decoding algorithm with PSI for Δ-linearly inter-dependent sources. We first consider a special case where $\Delta = 2^k$ and then generalize the case to $\Delta \neq 2^k$.

Property 2. *For Δ-linearly inter-dependent sources, the expected distortion of approximate decoding with $\Delta = 2^k$ $(0 \le k < M)$ is always lower than that with Δ' such that $2^{k-1} < \Delta' < 2^{k+1}, \Delta' \neq 2^k$.*

Proof. Any element in $GF(2^M)$ can be represented by M-bit binary numbers. Since $\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta \cdot \mathbf{1}$ in (16), there is no distortion if

$$(\mathbf{x}_t + \Delta \cdot \mathbf{1}) \oplus (\Delta \cdot \mathbf{1}) = \mathbf{x}_t. \tag{18}$$

Let $\omega(\mathbf{x}, \mathbf{y})$ denote the number of "1" that are at the same position of binary representations of $\mathbf{x}$ and $\mathbf{y}$. Note that, the condition given in (18) is satisfied if and only if $\omega(\mathbf{x}_t, \Delta \cdot \mathbf{1}) = 0$, because an overlap between any two "1"s at the same position of $\mathbf{x}_t$ and $\Delta \cdot \mathbf{1}$ results in carriage returns from $x_t^{(i)} + \Delta$. This leads to decoding errors (or distortion), implying that the distortion can be minimized with the smallest $\omega(\mathbf{x}_t, \Delta \cdot \mathbf{1})$. Since

$$\min \omega\left(x_t^{(i)}, \Delta\right) = 1$$

for all i, the distortion is minimized when $\Delta = 2^k$, $0 \le k < M$. This is equivalent to the case where only a single bit 1 is set at the $(k+1)$-th position of the binary representation of Δ, which completes the proof.□

In order to construct the PSI, the candidates $\boldsymbol{\Delta}_t$ need to be found, as discussed in Property 3. To be consistent with the assumptions on the sources, $x_t^{(i)} + \Delta$ does not exceed the size of GF, i.e., $0 \le x_t^{(i)} < 2^M - 2^k$ such that all source data are in $GF(2^M)$.

Property 3. *If $\Delta = 2^k$ $(0 \le k < M)$ in $GF(2^M)$, there are at most $(M-k)$ candidates in $\boldsymbol{\Delta}_t$ where n-th candidate Δ_n is expressed as*

$$\Delta_n = \sum_{i=k}^{k+n-1} 2^i$$

with the probability of

$$\Pr(\Delta_R = \Delta_n) = \frac{2^{(M-k-n)}}{2^{(M-k)} - 1}.$$

Proof. An element in $GF(2^M)$ is represented by M bits. For an element $\Delta = 2^k$ in $GF(2^M)$, it can be represented by only a single bit "1" at the $(k+1)$-th position and "0" at all the other positions. Recall that the candidates are generated when

$$x_{t+1}^{(i)} \oplus \Delta = \left(x_t^{(i)} + \Delta\right) \oplus \Delta \neq x_t^{(i)}.$$

If bit "1" is set for $(k+1)$-th, …, $(k+n-1)$-th position of $x_t^{(i)}$ when the bit "1" is set for both $(k+1)$-th position of Δ and $x_t^{(i)}$, Δ_n is generated with a value of $\Delta_n = \sum_{i=k}^{k+n-1} 2^i$. Since Δ_n is in $GF(2^M)$, $1 < n \le M-k$, there are at most $M-k$ candidates.

Next, we show that $\Pr(\Delta_R = \Delta_n) = 2^{(M-k-n)}/(2^{(M-k)} - 1)$. Since $x_t^{(i)}$ is in the range of $0 \le x_t^{(i)} < 2^M - 2^k$, $x_t^{(i)}$ can take one of $2^M - 2^k$ different values. Given a Δ_n, there are 2^{M-n} different values for $x_t^{(i)}$ because the $(k+n)$-th position of $x_t^{(i)}$ is set to "0" and the positions of $x_t^{(i)}$ from $(k+n-1)$-th to $(k+1)$-th are set to "1". Therefore, Δ_n is generated with probability

$$\Pr(\Delta_R = \Delta_n) = \frac{2^{(M-n)}}{2^M - 2^k} \tag{19}$$

Table 3
Q5 Examples of candidates for $\Delta = 2^k$ over $GF(2^8)$.

	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6
$\Delta = 2^2$	2^2	$\sum_{k=2}^{3} 2^k$	$\sum_{k=2}^{4} 2^k$	$\sum_{k=2}^{5} 2^k$	$\sum_{k=2}^{6} 2^k$	$\sum_{k=2}^{7} 2^k$
$\Delta = 2^3$	2^3	$\sum_{k=3}^{4} 2^k$	$\sum_{k=3}^{5} 2^k$	$\sum_{k=3}^{6} 2^k$	$\sum_{k=3}^{7} 2^k$	
$\Delta = 2^4$	2^4	$\sum_{k=4}^{5} 2^k$	$\sum_{k=4}^{6} 2^k$	$\sum_{k=4}^{7} 2^k$		
$\Delta = 2^5$	2^5	$\sum_{k=5}^{6} 2^k$	$\sum_{k=5}^{7} 2^k$			
$\Delta = 2^6$	2^6	$\sum_{k=6}^{7} 2^k$				

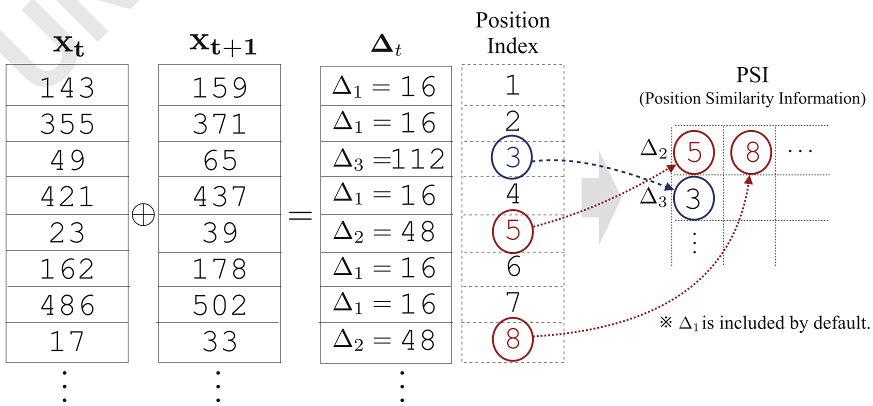

Fig. 5. An illustrative example of PSI construction. In this example, $\mathbf{x}_t - \mathbf{x}_{t+1} = \Delta = 16$ in $\mathbb{R}$. However, $\mathbf{x}_t \oplus \mathbf{x}_{t+1}$ in GF includes not only 16 but also other values such as 48, 112, …, in $GF(2^8)$. The results from $\mathbf{x}_t \oplus \mathbf{x}_{t+1}$ are referred to as *candidates* of $\Delta = 16$. The PSI includes Δ_1 by default and the position indices for Δ_n for $n \ge 2$.

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

$$\Pr(\Delta_R = \Delta_n) = \frac{2^{(M-k-n)}}{2^{(M-k)}-1} \quad \text{for } 1 \le n \le M-k \tag{20}$$

which completes the proof.□

The candidates $\mathbf{\Delta}_t$ and the corresponding probability mass function (PMF) based on Property 3 are shown in Table 4 when $\Delta = 2^k$ in $GF(2^M)$. This enables us to construct a PSI matrix, where each row of PSI represents a candidate. For example, the first row has the index numbers of Δ_2, the second row has the index numbers of Δ_3, and in general, the $(M-n-1)$-th row has the index numbers of Δ_{M-n} for $n = 1, \ldots, k$. As discussed above, while the approximate decoding with the PSI leads to improved performance, the use of PSI results in more communication overhead compared to SI. Hence, it is essential to investigate the tradeoff between the amount of overhead for transmitting PSI and the performance improvement. This tradeoff is studied in Property 4.

Property 4. *If the $(n-1)$-th row of PSI or Δ_n is additionally used for approximate decoding, its performance can be improved by a factor of $1/2^{(n-1)}$.*

Proof. As shown in Table 4, $\Pr(\Delta_R = \Delta_n)$ is given by $2^{(M-k-n)}/2^{(M-k)}-1$. This means that $\Pr(\Delta_R = \Delta_n)$ decreases by $1/2^n$, as n increases. Thus, if Δ_n is additionally used for approximate decoding, the performance can be improved by $1/2^{(n-1)}$ times compared to the gain obtained if Δ_{n-1} is used.□

The performance improvement based on additional information in PSI is interpreted as an *additional refinement*. Property 4 is used as a guideline for the trade-off between the communication overhead and the decoding performance of the proposed approximate decoding algorithm.

Unlike the case of $\Delta = 2^k$, the candidates for the case of $\Delta \ne 2^k$ cannot be easily identified. In this case, the candidates $\mathbf{\Delta}_t$ can be found by decomposing into the sum of 2^ks for $0 \le k \le M$ and using the candidates of $\Delta = 2^k$. The above description is explained in Algorithm 1.

Algorithm 1. Candidates $\mathbf{\Delta}_t$.

Given: Δ and $GF(2^M)$.
if $\Delta = 2^k$ **then**
  $\mathbf{\Delta}_t = \{\Delta_1, \ldots, \Delta_n, \ldots, \Delta_{M-k}\}$ where $\Delta_n = \sum_{i=k}^{k+n-1} 2^i$ // `Property 3`
else // `if` $\Delta \ne 2^k$
  Find $p_1, p_2, \ldots, p_n$ such that $\Delta = \sum_{j=1}^{n} 2^{p_j}$ for $0 \le p_n < \cdots < p_2 < p_1$
  Find all the candidates $\mathbf{\Delta}_t$ based on Property 3, if $\Delta = 2^{p_n}$ (denoted by $\mathbf{\Delta}_t(2^{p_n})$).
  Compute Cartesian product set, $\prod_{l=1}^{n} \mathbf{\Delta}_t(2^{p_l}) = \{(\gamma_1, \ldots, \gamma_n) \mid \gamma_i \in \mathbf{\Delta}_t(2^{p_i})\}$
  Calculate the sum of a tuple $\Gamma_i = \sum_{j=1}^{n} \gamma_j$, where Γ_i is sum of i-th tuple elements
  $S = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_q\}$, where $q = \prod_{l=1}^{n} |\mathbf{\Delta}_t(2^{p_l})|$ and $|\cdot|$ denotes cardinality
  if there are even number of Γ_i in S **then**
    Remove Γ_i from S
  else // `odd numbers`
    if there are multiple instances of Γ_i in S **then** merge them to a single Γ_i
    end if
  $\mathbf{\Delta}_t \leftarrow S$
  end if
end if

4.3. PSI properties illustrations

We implement here our approximate decoding method in conjunction with PSI and we experimentally analyze the properties studied in the previous section. In the following simulations, we consider a set of three Δ-linearly inter-dependent sources, which are randomly generated as in (16).

Fig. 6 shows the performance of approximate decoding with PSI, which is measured by PMSE over various packet loss rates with different amounts of additional information delivered by the PSI.

In these experiments, the first source data vector $\mathbf{x}_1$ is randomly generated and neighbor source vectors are generated by $\mathbf{x}_t = \mathbf{x}_1 + (t-1) \cdot \Delta$ for $t = 1, \ldots, T$. Hence, the sources are Δ-linearly inter-dependent. For Fig. 6(a) and (b), $\Delta = 8$ and $\Delta = 10$ are used, respectively. Since $\Delta = 10$ is not the form of 2^k, candidates are found based on Algorithm 1. Ten data vectors are combined together based on RLNC in $GF(2^{10})$. The experiments are independently repeated 1000 times.

It is clear from Fig. 6(a) and (b) that the approximate decoding solution for different PSI outperforms the approximate decoding with PSI_0 or SI, over all the range of packet loss rates. Moreover, the approach proposed in Algorithm 1 is effective for any Δs that are not necessarily powers of 2. The results indicate that the additional information given by PSI needs to be adaptively structured by explicitly considering the packet loss rates and the targeted performance.

Fig. 7 confirms Property 4, which says that additional Δ_n can improve the performance of approximate decoding by a factor of $1/2^{n-1}$. For example, if Δ_2 is used in addition to Δ_1, the performance (measured by PMSE) improves from approximately 0.06 to 0.03. Similarly, if Δ_3 is used in addition to Δ_1 and Δ_2, the performance improves again from approximately 0.03 to 0.015. We finally note that there is no error if $\Delta_1, \ldots, \Delta_6$ are included in PSI. This is because $\Delta_1, \ldots, \Delta_6$ are the information that can be maximally provided to the decoder (recall that $M-k = 8-2 = 6$ for $\Delta = 2^k = 4$ and $GF(2^M) = GF(2^8)$ as discussed in Property 3).

Table 4
Candidates of Δ and PMF of the candidates if $\Delta = 2^k$ in $GF(2^M)$.

	Δ_1	Δ_2	⋯	Δ_n	⋯	Δ_{M-k}
$\mathbf{\Delta}_t$	2^k	$\sum_{j=k}^{k+1} 2^j$	⋯	$\sum_{j=k}^{k+(n-1)} 2^j$	⋯	$\sum_{j=k}^{M-1} 2^j$
$\Pr(\Delta_R = \Delta_n)$	$\frac{2^{(M-k-1)}}{2^{(M-k)}-1}$	$\frac{2^{(M-k-2)}}{2^{(M-k)}-1}$	⋯	$\frac{2^{(M-k-n)}}{2^{(M-k)}-1}$	⋯	$\frac{1}{2^{(M-k)}-1}$

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

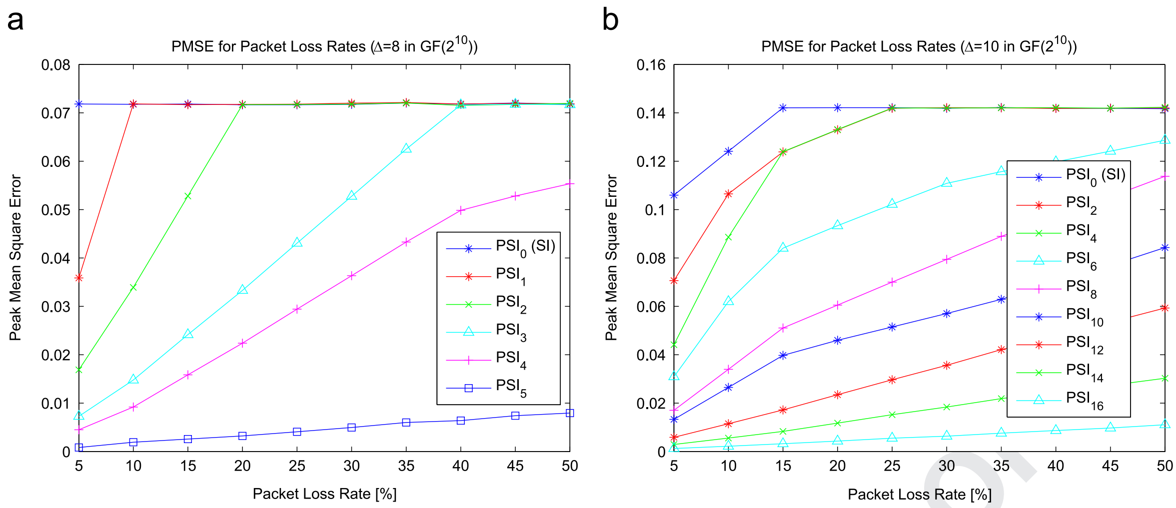

Fig. 6. Average PMSE of the proposed decoding algorithm for various sizes of PSI. In order to find candidates for $\Delta=10$, Algorithm 1 is used. PSI_0 denotes the case where only SI is delivered and PSI_n denote cases where $\{\Delta_1,\ldots,\Delta_n\}$ is included in PSI. (a) $\Delta=8$, $GF(2^{10})$ and (b) $\Delta=10$, $GF(2^{10})$.

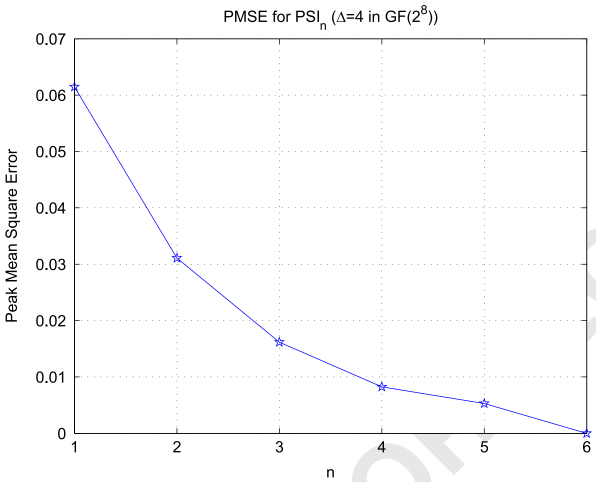

Fig. 7. Average PMSE for PSI_n. PSI_n $(1\le n\le 6)$ denote cases where $\{\Delta_1,\ldots,\Delta_n\}$ is contained in PSI.

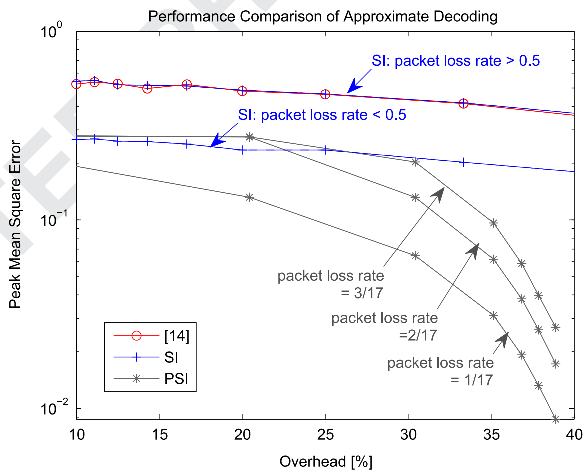

Fig. 8. Performance comparison for three different approximate decoding algorithms.

5. Simulation results

In this section, we quantitatively evaluate the performance of the proposed approaches and compare them with other existing approaches such as the approach presented in [14], Strategy 1 (S1) discussed in Section 3.2 (denoted by the *SI scheme*), and the method described by (17) (denoted by the *PSI scheme*). We then deploy the proposed approach in a sensor network scenario.

5.1. Performance comparison of approximate decoding algorithms

Unlike the approaches shown in [14] and the SI scheme, the proposed PSI scheme requires additional information as a form of PSI, which can be considered as an overhead. The overhead is defined in the simulations as the ratio of the amount of data additionally added to the amount of original data. For fair comparison, the source node in [14] and the SI scheme repeatedly transmit packets such that they have the same amount of overhead as the PSI scheme, since [14] and the SI scheme do not require additional information for decoding. For example, 25% overhead for [14] and the SI scheme means that one packet is transmitted in addition to every four packets, resulting in five transmissions in total. In case of the PSI scheme, however, the amount of additional information included in PSI is 25% larger than the total packet size (i.e., four packets).

The simulation results for performance comparison are shown in Fig. 8. In the simulations, $\Delta=8$ and 17 packets are network coded over $GF(2^{10})$. Fig. 8 shows the average PMSE for the considered approaches over 100 independent experiments. For the SI scheme, the packet loss rates are set in the range of (0, 0.5] and (0.5, 1), and the corresponding average PMSEs shown in Fig. 8 are indicated by “SI: packet loss rate <0.5” and “SI: packet loss rate >0.5”, respectively. For the PSI scheme, the packet loss rates are set to 1/17, 2/17 and 3/17 for illustration.

It can be easily observed that the performance of all the approaches improves as the overhead increases. This is because the additional information can help the decoder to recover more data correctly. In particular, the impact of the additional information on the performance improvement of the PSI scheme is the greatest among the considered approaches, which is consistent with the analytical results discussed in Section 3.2. It is also observed that the performance curves converge into two PMSE levels as the overhead approaches 0%, which are determined based on the packet loss rates, i.e., packet loss rate is higher or lower than 50%. If the packet loss rate is higher than 50%, approximate decoding needs more *artificially* generated information (e.g., using (4)) than innovative (received) packets, thereby leading to performance degradation.

5.2. Illustrative application: approximate decoding deployment in sensor networks

In this section, we consider an illustrative application of the proposed approximate decoding algorithm with PSI in seismographic networks. In this simulation, we use seismic data that is actually collected from 30 sensors that measure the amplitude of a seismic signal at a distance of 100 m from each other. A sensor t captures signal $\mathbf{s}_t$ that represents a series of sampled observations in a time window of size L, i.e., $\mathbf{s}_t = [s_t^{(1)}, \ldots, s_t^{(L)}]^T$. The measured symbol is quantized and mapped into $GF(2^M)$ elements, and the resulting data is denoted by x_t, i.e., $\mathcal{Q}(s_t^{(i)}) = x_t^{(i)} \in GF(2^M)$. Each sensor makes measurements and forwards them to its neighbor encoding sensors. An index vector of sensors whose data are encoded together at sensor t is denoted by $\mathbf{H}_i \in t$, where i is a vector index. An encoding sensor t may receive multiple data from sensors in $\mathbf{H}_i$ and combine them based on RLNC, where the combined data is again forwarded to its neighbor nodes or the final destination (i.e., receiver). The elements of coding coefficient vector $\mathbf{c}_t$ are randomly selected from $GF(2^M)$. The encoded data packets generated at the first coding stage are expressed as $\mathbf{y}^{(i)}(2) = \sum_{t \in \mathbf{H}_i} \oplus\, \mathbf{c}_t(1) \otimes x_t^{(i)}$. The encoded packets are mixed again at the intermediate nodes while traversing network towards the destination. An illustrative example for simulation setup is shown in Fig. 9.

As a representative example, we consider the case where the data measured from sensors 1–3 are collected and combined by sensor 3 based on RLNC in our experiments. The measured signals from a set of sensors are mostly time-shifted and energy-scaled. The energy difference between sensors is modeled as the Gaussian distribution. The similarity of the signals becomes higher as sensors are closer to each other. In this simulation, the energy difference between the signals captured by neighbor sensors is modeled as Gaussian random variable with mean value of 8 and variance of 0.2^2, i.e., $s_{t+1}^{(i)} - s_t^{(i)} \sim \mathcal{N}(8, 0.2^2)$ for $t = 1, 2$. In Fig. 10, the seismic data measured from three sensors are depicted. We set the temporal window size by $L = 256$ for data representation and use the quantizer $\mathbf{x}_t \in GF(2^M)$ for source data quantization. The *GF* size is set to 2^7, i.e., $GF(2^7)$ and packet loss rate is set to 1/3. The receiver reconstructs the original source data based on the data collected from the three sensors.

In the decoding process, we deploy SI as $\boldsymbol{\nu}$, i.e., $\boldsymbol{\nu} = 8 \cdot \mathbf{1}_{(T-K_{in})}$, in (4). This is the S1 discussed in Section 3.2. We also deploy the proposed decoding algorithm including the PSI, which is discussed in Section 4. The results are shown in Fig. 11. Fig. 11(a-1) shows the original signal captured by sensor 3 and Fig. 11(a-2) shows the decoded signal based on S1 that depends only on SI. Fig. 11(a-3) and (a-4) shows the decoded signals based on PSI with Δ_2 and with Δ_2 and Δ_3, respectively. The average PMSE is presented in Table 5, where the amount of decoding error is

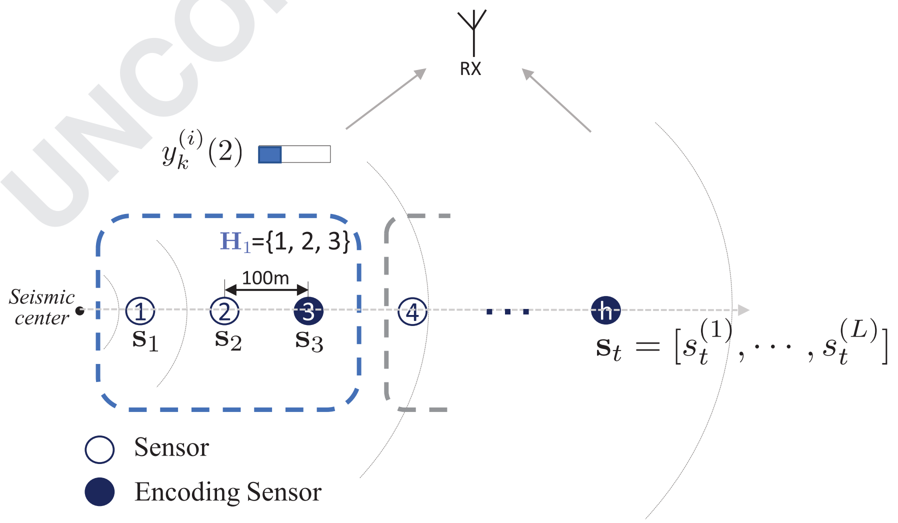

Fig. 9. An illustrative example of sensor network for seismic data. Sensors are deployed and capture seismic signals. The symbols $s_1^{(i)}, s_2^{(i)}$, and $s_3^{(i)}$ measured at sensors 1, 2 and 3, respectively, are transformed into *GF* (i.e., $x_1^{(i)}, x_2^{(i)}, x_3^{(i)}$) and encoded based on RLNC. In this example, $y_k^{(i)}(2) = \{c_{k1}(1) \otimes x_1^{(i)}\} \oplus \{c_{k2}(1) \otimes x_2^{(i)}\} \oplus \{c_{k3}(1) \otimes x_3^{(i)}\}$. The encoded data forwarded to their neighbor nodes through error-prone networks. Some packets can be lost in the transmission and the receiver reconstructs source data based on the proposed approximate decoding algorithm.

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

reduced to 17.3% and 6.7% of (a-2), in (a-3) and (a-4), respectively. As discussed, the performance of approximate decoding improves as more Δ_i is included in PSI.

6. Related works

In this section, we study prior work related with the proposed approaches. Several regularization techniques [28] can be deployed for overcoming the all-or-nothing problem of network coding in finite fields. For example, the pseudo-inverse of underdetermined coefficient matrix

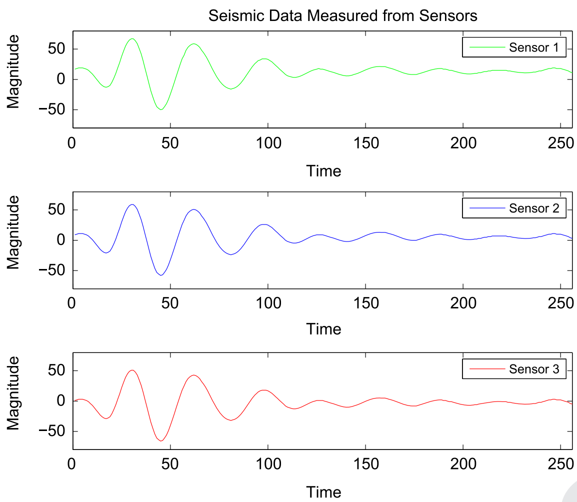

Fig. 10. Seismic data measured from sensors. Sensor 1 measures the seismic data and forwards it to sensor 2 with energy attenuation with $\mathcal{N}(8, 0.2^2)$.

can be used for decoding. However, it is generally known that this type of regularization techniques may result in unreasonable approximations [28]. Alternatively, Tikhonov regularization [29] can improve the decoding performance by slightly modifying the standard least square approach. However, this technique cannot be easily deployed in practice because it requires to determine additional optimization parameters. Sparsity assumptions can also be used for regularized decoding in underdetermined systems in cases where a model of the signal of interest is known a priori [30]. However, all of these regularization techniques have been designed and developed in the field of real numbers, but not in finite fields that are used in network coding approaches. Hence, they may show significant performance degradation if they are blindly deployed in the proposed framework, as they cannot consider several properties (e.g., cyclic properties) of finite field operations. Underdetermined systems can also be solved approximately based on the maximum likelihood estimation (MLE) techniques (see e.g., Part II of [31]) or based on mixed integer linear programming [32]. However, these techniques require effective data models and may typically involve large computational complexity.

Unlike the approaches mentioned above, several approaches have been proposed to overcome the all-or-nothing problem of network coding in finite fields (see e.g., [33,32]). In [33], a loss-tolerant protocol for broadcasting, named by Dragoncast, based on network coding is proposed. In [32], a coding scheme that jointly considers both network coding and multiple description coding is proposed. The problem of data reconstruction is formulated as a mixed integer quadratic programming and robustness against missing packets is achieved. Our prior work [14] propose an approximate decoding approach based on

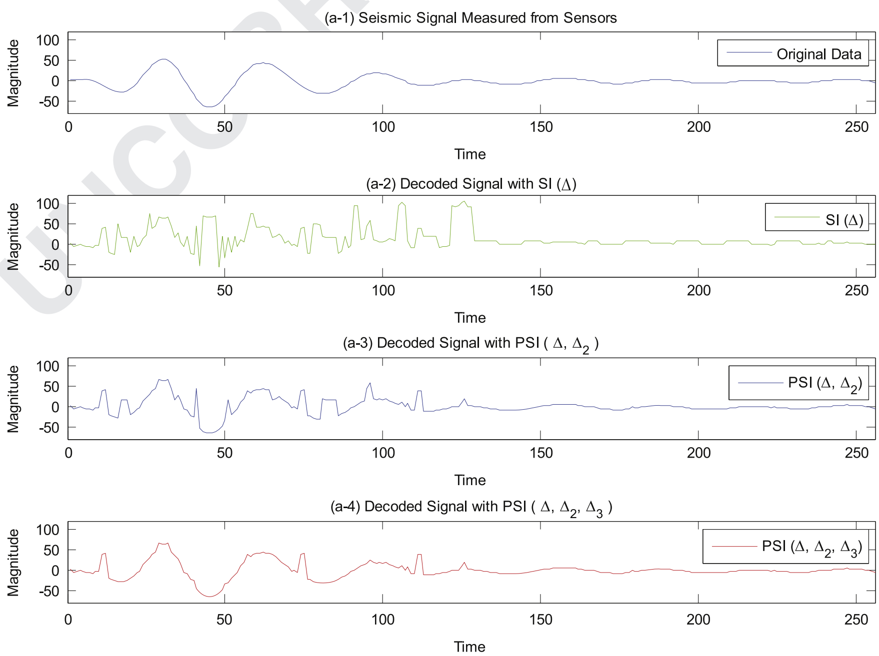

Fig. 11. Seismic signals measured from sensors and decoded signals by approximate decoding with PSI. Energy difference on the signals is modeled by Gaussian distribution, $\mathcal{N}(8, 0.2^2)$, and network coding and approximate decoding are performed in $GF(2^7)$.

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

Table 5
Average PMSE for different approximate decoding algorithms.

	Fig. 11 (a-2) SI (Δ)	Fig. 11 (a-3) PSI (Δ, Δ_2)	Fig. 11 (a-4) PSI ($\Delta, \Delta_2, \Delta_3$)
PMSE(%)	0.0010 (100 %)	1.7266×10^{-4}(17.3%)	6.7118×10^{-5}(6.7%)

simple matches of the most similar data between neighbor sources.

The information on the inter-dependency of the sources has been exploited in order to improve the performance of data reconstruction. In [34], network coding for two arbitrary inter-dependent sources is studied in a generic network and an upper bound on the probability of decoding errors is found as a function of the network parameters such as the number of links upstream of a receiver or network topologies. Several practical aspects of joint source and network coding are investigated in [35]. However, the proposed approach is sub-optimal and provides a solution only for two inter-dependent sources that are transmitted over binary symmetric channels. The design of optimal network codes under joint distributed source and network coding framework is studied in a sensor network [36]. The goal is to find the optimal tradeoff between compression efficiency and network robustness. However, the complexity of the proposed solution in [36] grows exponentially with the network size. In [23], an iterative decoding algorithm for source reconstruction is proposed for inter-dependent sources following a belief propagation approach that incorporates inter-dependence characteristics. However, none of prior works solve the all-or-nothing problem of network coding in finite fields with low-complexity, while the algorithm proposed in this paper can be implemented with low-complexity and it is compatible with well-known Gaussian elimination methods. Several extensions of [14] can be found in [15] and [16] for deterministic sources with SI and PSI, respectively. In [17], approximate decoding algorithm for sources that have bell-shaped, symmetric and unimodal distribution is proposed. Unlike these prior works, we propose more generalized framework that can cover a large range of source types. Moreover, the proposed approximate decoding solution can exploit both SI and PSI such that the approximate decoding performance can be significantly improved.

Since additional information is transmitted [37] in the proposed scenario, the proposed approach can be considered as an index coding problem [38]. However, the index coding is generally considered in broadcasting scenarios (i.e., a single server wishes to communicate with several clients) and the side information in the index coding contains the source information [39]. Hence, the amount of additional information required for the proposed approach is significantly smaller than that for index coding.

7. Conclusion

In this paper, we consider the transmission of network coded inter-dependent sources in error prone networks. In order to solve the all-or-nothing problem of network coding approaches, we propose a solution that approximately reconstructs the source data when the number of received data packets is not sufficient for perfect recovery without major change in the conventional decoding architecture. Unlike prior works, we consider a generalized source inter-dependency characterized by unimodal shapes. Given the information about source distributions, we show that the performance of the proposed approximate decoding algorithm is improved by using the mode of the distributions (SI). We further improve the performance of the proposed approximate decoding algorithms by deploying additional information called PSI. We evaluate the proposed approaches in an illustrative example of sensor network and the simulation results confirm that approximate decoding along with minimal side information leads to effective source reconstruction.

Appendix A

In Appendix A, we show that the approximate decoding performance is maximized only when $\nu_n = \nu_n^*$ and is flat for all other value of $\nu_n (\neq \nu_n^*)$ values, which justify the use of the Cauchy–Dirac delta function in Section 3.2.

The approximate decoding algorithm in (4) can be expressed as

$$\hat{\mathbf{x}}^{(i)} = \begin{bmatrix} \mathbf{C}(h_D-1) \\ \mathbf{D} \end{bmatrix}^{-1} \odot \begin{bmatrix} \mathbf{y}^{(i)}(h_D) \\ \boldsymbol{\nu} \end{bmatrix} = \overline{\mathbf{C}} \odot \begin{bmatrix} \mathbf{y}^{(i)}(h_D) \\ \boldsymbol{\nu} \end{bmatrix} = [\overline{\mathbf{c}}_1, \ldots, \overline{\mathbf{c}}_{K_{in}}, \overline{\mathbf{c}}_{K_{in}+1}, \ldots, \overline{\mathbf{c}}_T] \odot \begin{bmatrix} \mathbf{y}^{(i)}(h_D) \\ \boldsymbol{\nu} \end{bmatrix} = [\overline{\mathbf{c}}_1, \ldots, \overline{\mathbf{c}}_{K_{in}}] \odot \mathbf{y}^{(i)}(h_D) \oplus [\overline{\mathbf{c}}_{K_{in}+1}, \ldots, \overline{\mathbf{c}}_T] \odot \boldsymbol{\nu}$$

where $\overline{\mathbf{C}} \triangleq \left[\mathbf{C}(h_D-1)^T, \mathbf{D}^T\right]^{-T} = [\overline{\mathbf{c}}_1, \ldots, \overline{\mathbf{c}}_T]$ and $\overline{\mathbf{c}}_t$ is in $GF(2^M)$. Hence, the squared decoding error can be expressed as

$$\left|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\right|^2 = \left|[\overline{\mathbf{c}}_{K_{in}+1}, \ldots, \overline{\mathbf{c}}_T] \odot (\boldsymbol{\nu}^{*T} - \boldsymbol{\nu}^T)\right|^2. \quad (21)$$

The squared decoding error shown in (21) implies that $\boldsymbol{\nu} = \boldsymbol{\nu}^*$ is the condition that minimizes the error. However, a small distance between $\boldsymbol{\nu}$ and $\boldsymbol{\nu}^*$ ($\boldsymbol{\nu} \neq \boldsymbol{\nu}^*$) in the GF does not directly lead to small distance of $\left|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\right|^2$ due to the cyclic property of the operations in the GF and the random selection of the coding coefficients. The characteristics of the decoding performance (i.e., the squared decoding error) can thus be well captured by the Cauchy–Dirac delta function. An illustrative example shown in Fig. 12 for the squared decoding error in (21) also confirms the discussion above.

Please cite this article as: M. Kwon, et al., Approximate decoding for network coded inter-dependent data, Signal Processing (2015), http://dx.doi.org/10.1016/j.sigpro.2015.09.010

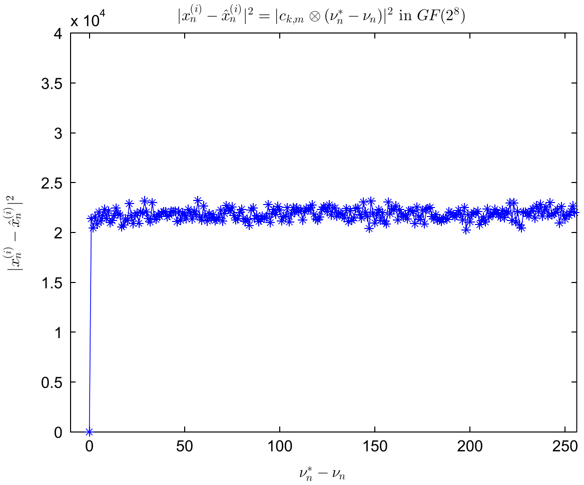

Fig. 12. An illustrative example of the squared decoding error $|x_n^{(i)} - \hat{x}_n^{(i)}|^2 = |c_{k,m} \otimes (\nu_n^* - \nu_n)|^2$ in $GF(2^8)$. $c_{k,m}$ is randomly selected in $GF(2^8)$.

References

[1] R. Ahlswede, N. Cai, S.-Y.R. Li, R.W. Yeung, Network information flow, IEEE Trans. Inf. Theory 46 (July (4)) (2000) 1204–1216.

[2] S.-Y.R. Li, R.W. Yeung, N. Cai, Linear network coding, IEEE Trans. Inf. Theory 49 (February (2)) (2003) 371–381.

[3] Z. Li, B. Li, D. Jiang, L.C. Lau, On achieving optimal throughput with network coding, in: IEEE International Conference on Computer and Communications (INFOCOM), vol. 3, Miami, FL, USA, March 2005, pp. 2184–2194.

[4] P.A. Chou, Y. Wu, Network coding for the internet and wireless networks, IEEE Signal Process. Mag. 24 (September (5)) (2007) 77–85.

[5] T. Ho, R. Koetter, M.Médard, D. Karger, M. Effros, The benefits of coding over routing in a randomized setting, in: IEEE International Symposium on Information Theory, Cambridge, MA, USA, June–July 2003.

[6] C. Gkantsidis, P.R. Rodriguez, Network coding for large scale content distribution, in: IEEE International Conference on Computer and Communications (INFOCOM), vol. 4, March 2005, pp. 2235–2245.

[7] S. Acedański, S. Deb, M. Médard, R. Koetter, How good is random linear coding based distributed networked storage, in: Workshop on Network Coding, Theory and Applications, Riva del Garda, Italy, April 2005.

[8] S. Deb, M. Médard, C. Choute, On random network coding based information dissemination, in: IEEE International Symposium on Information Theory, Adelaide, Australia, September 2005, pp. 278–282.

[9] C. Gkantsidis, J. Miller, P. Rodriguez, Comprehensive view of a live network coding P2P system, in: ACM SIGCOMM Conference on Internet Measurement, Brazil, October 2006.

[10] A.G. Dimakis, P.B. Godfrey, M.J. Wainwright, K. Ramchandran, Network coding for distributed storage systems, in: IEEE International Conference on Computer and Communications (INFOCOM), Anchorage, AK, USA, May 2007.

[11] T. Ho, M. Médard, J. Shi, M. Effros, D.R. Karger, On randomized network coding, in: Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, October 2003.

[12] M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, B. Leong, A random linear network coding approach to multicast, IEEE Trans. Inf. Theory 52 (October (10)) (2006) 4413–4430.

[13] C. Fragouli, E. Soljanin, Network coding applications, Found. Trends Netw. 2 (2007).

[14] H. Park, N. Thomos, P. Frossard, Approximate decoding approaches for network coded correlated data, Signal Process. 93 (January (1)) (2013) 109–213.

[15] M. Kwon, H. Park, An improved approximate decoding with correlated sources, in: SPIE Optical Engineering+ Applications. International Society for Optics and Photonics, San Diego, CA, USA, August 2011.

[16] M. Kwon, H. Park, P. Frossard, Improved approximate decoding based on position information matrix, in: IEEE Symposium on Computers and Communications, Cappadocia, Turkey, July 2012.

[17] M. Kwon, H. Park, Approximate recovery of network coded real-time information, in: International Conference on Information Networking (ICOIN), Phuket, Thailand, February 2014, pp. 545–549.

[18] K.-J. Bathe, E.L. Wilson, Numerical Methods in Finite Element Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1976.

[19] P. Paweczak, S. Pollin, H.W. So, A. Bahai, R. Prasad, R. Hekmat, Comparison of opportunistic spectrum multichannel medium access control protocols, in: IEEE Global Telecommunications Conference (GLOBECOM2008), 2008, pp. 1–6.

[20] B.F. Lo, A survey of common control channel design in cognitive radio networks, Phys. Commun. 4 (1) (2011) 26–39.

[21] K. Bian, J. Park, R. Chen, Control channel establishment in cognitive radio networks using channel hopping, IEEE J. Sel. Areas Commun. 29 (4) (2011) 689–703.

[22] L. Lazos, S. Liu, M. Krunz, Mitigating control-channel jamming attacks in multi-channel ad hoc networks, in: Proceedings of the Second ACM Conference on Wireless Network Security, 2009, pp. 169–180.

[23] E. Bourtsoulatze, N. Thomos, P. Frossard, Correlation-aware reconstruction of network coded sources, in: IEEE International Symposium on Network Coding, Cambridge, MA, USA, June 2012.

[24] J. Liu, Q. Yang, B. Yao, R. Brown, G. Yue, Linear correlation between fractal dimension of eeg signal and handgrip force, Biol. Cybern. 93 (2) (2005) 131–140.

[25] R. Bacastow, C. Keeling, T. Whorf, Seasonal amplitude increase in atmospheric CO_2 concentration at Mauna Loa, Hawaii, 1959–1982, J. Geophys. Res.: Atmos. (1984–2012) 90 (D6) (1985) 10529–10540.

[26] A.M. Eskicioglu, P.S. Fisher, Image quality measures and their performance, IEEE Trans. Commun. 43 (12) (1995) 2959–2965.

[27] M.G. Katz, D. Tall, A Cauchy–Dirac delta function, Found. Sci. 18 (1) (2013) 107–123.

[28] A. Neumaier, Solving ill-conditioned and singular linear systems: a tutorial on regularization, SIAM Rev. 40 (September (3)) (1998) 636–666.

[29] G.H. Golub, P.C. Hansen, D.P. O'Leary, Tikhonov regularization and total least squares, SIAM J. Matrix Anal. Appl. 21 (1) (1999) 185–194.

[30] E.J. Candès, M.B. Wakin, An introduction to compressive sampling, IEEE Signal Process. Mag. 25 (March (2)) (2008) 21–30.

[31] S. Bhattacharyya, Handbook of Signal Processing Systems, Springer, 2010. Q4

[32] L. Iwaza, M. Kieffer, L. Liberti, K. Al-Agha, Joint decoding of multiple-description network-coded data, in: International Symposium on Network Coding, Beijing, China, July 2011.

[33] S.Y. Cho, C. Adjih, Wireless broadcast with network coding in mobile ad-hoc networks: Dragoncast, in: Proceedings of the EU-Korea Conference on Science and Technology, vol. 124, 2008, pp. 333–344.

[34] T. Ho, M. Médard, M. Effros, R. Koetter, Network coding for correlated sources, in: Conference on Information Sciences and Systems, Princeton, NJ, USA, March 2004.

[35] Y. Wu, V. Stankovic, Z. Xiong, S. Y. Kung, On practical design for joint distributed source and network coding, in Workshop on Network Coding, Theory, and Applications, Riva del Garda, Italy, April 2005.

[36] X. Zhang, S.B. Wicker, Robustness vs. efficiency in sensor network, in: International Symposium on Information Processing in Sensor Networks, Los Angeles, CA, USA, April 2005.

[37] Z. Bar-Yossef, Y. Birk, T.S. Jayram, T. Kol, Index coding with side information, IEEE Trans. Inf. Theory 57 (March 3) (2011) 1479–1494.

[38] Y. Birk, T. Kol, Informed-source coding-on-demand (iscod) over broadcast channels, in: Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 1998), vol. 3, March 1998, pp. 1257–1264.

[39] M. Effros, S. El Rouayheb, M. Langberg, An equivalence between network coding and index coding, in: IEEE International Symposium on Information Theory Proceedings (ISIT 2013), July 2013, pp. 967–971.