Improved Approximate Decoding based on Position Information Matrix

Minhae Kwon, Hyunggon Park
Multimedia Communications and Networking Laboratory
Ewha Womans University, Seoul, Korea
kyon0516@ewhain.net; hyunggon.park@ewha.ac.kr

Pascal Frossard
Signal Processing Laboratory (LTS4)
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
pascal.frossard@epfl.ch

***Abstract*—This paper proposes a robust decoding algorithm in delivery of network coded data which is in particular correlated and delay-sensitive. We consider ad-hoc sensor network topologies, where a correlated data is delivered based on network coding techniques in conjunction with approximate decoding algorithm in order for efficient and robust data delivery. The approximate decoding algorithm has been developed as a decoding solution to ill-posed problems for network coded correlated data sources. In this paper, we improve the performance of approximate decoding algorithm by explicitly considering more information, which is used to additionally refine the recovered data. The information includes potential results that are from finite field operations and the set of such information is referred to as position information matrix in this paper. We deploy the position information matrix into approximate decoding algorithm and investigate its corresponding properties. We then analytically show that this improves the performance of approximate decoding algorithm. Our simulation results confirm the properties of the proposed approximate decoding algorithm with position information matrix and improved performance.**

***Index Terms*—Network coding, approximate decoding, position information matrix (PIM), correlated source data, distributed delivery, ad hoc sensor networks.**

I. Introduction

Ad-hoc sensor network becomes widely deployed for monitoring geospatial information such as humidity, temperature and soil fertility [1]. For real-time sensor network applications, studying the efficient solutions for information delivery in a timely manner is the one of main research issues. Due to the difficulty of centralized coordination, the information delivery among the sensors is typically performed in a distributed manner over ad-hoc topologies [2], [3].

Network coding [4] has been deployed as a method to build efficient distributed delivery algorithms in networks with path and source diversity. Network nodes in network coding are able to perform basic processing operations so that nodes combine information packets and forward the resulting data to the next network nodes. In [5], it is shown that network coding can improve the throughput of the system, achieve better max-flow min-cut limit of networks, and enhance the robustness to data loss. In this paper, we adopt random linear network coding (RLNC) [6], as RLNC based solutions in general can lead to negligible performance degradations compared to globally optimized solutions (e.g., linear network coding [5]), while requiring only low coordination between nodes, and the corresponding lower communication costs. In this paper, we consider the transmission of correlated data sources that are independently encoded based on RLNC at the sensors, transmitted over lossy ad hoc networks and jointly decoded in a timely manner.

Due to the dynamics of source and networks, there is no guarantee that each node receives enough packets for successful data recovery. It becomes critical for delay-sensitive applications because packets arriving after delay deadline are discarded, which incurs significant performance degradation. In order to solve this problem, approximate decoding algorithm has been introduced in [7], which enables recovery of the original source data even when the number of useful packets is not sufficient for perfect data reconstruction. However, the approximate algorithm in [7] exploits the correlation based on a simple best matching method, resulting in the limited performance of the approximate decoding. A more systematic model for data correlation is discussed in [8], where a linear correlation model is deployed with a measure of data similarity (i.e., similarity factor).

While the linear correlation model in [8] has improved approximate decoding algorithm by exploiting the source correlation in terms of a similarity factor, the unique characteristics of finite field operations is not fully considered, leading to the performance degradation. Thus, the goal of this paper is to further improve a prior solution discussed in [8] by considering more information such as unintended results from finite field operations. More specifically, we study what are the candidates that may be potentially generated from the finite field operations while taking into account the similarity factor. Then, we incorporate this information to refine the approximate decoding results, leading to a better performance. The set of such additional information is referred to as position information matrix (PIM). While such additional information can lead to an improved performance, it also requires higher communication cost. Hence, we investigate the tradeoff between the performance improvement and the amount of additional information.

This paper is organized as follows. In Section II, we

This research was supported in part by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program (NIPA-2012-H0301-12-1008, NIPA-2012-H0301-12-4004) supervised by the NIPA(National IT Industry Promotion Agency) and in part by by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2012-0002917).

978-1-4673-2713-8/12/$31.00 ©2012 IEEE

present our system setup and a brief overview of approximate decoding algorithm. In Section III, the position information matrix is introduced and several properties of the proposed algorithm in conjunction with the position information matrix are discussed. In Section IV, illustrative examples that confirm the investigated properties are presented. Finally, the conclusions are drawn in Section V.

II. System Setup

A. Framework of RLNC-based Data Transmission

In this paper, we consider an overlay ad hoc network, which consists of source, intermediate and client nodes. The goal is to deliver the source data to the client nodes through the intermediate nodes that are able to perform network coding operations. A similar framework can be found in [8].

N non-negative correlated source data are denoted by $x_1, x_2, \ldots, x_N$, where $x_n \in \mathcal{X}(\subset \mathbb{R})$ for $1 \le n \le N$. $\mathcal{X}$ is an alphabet set of x_n and $|\mathcal{X}|$ denotes the size of $\mathcal{X}$. Since network coding operations are performed in GF (Galois Field), each x_n can also be considered as an element in GF. In this paper, we use the superscript $\mathcal{F}$ in order to clearly express that elements are in GF. For example, $x_n^{\mathcal{F}}$ means that x_n is considered as an element in GF. This operation (and the corresponding inverse operation) is performed by an identity function, which maps a value in a field to the corresponding same value in another field (and vice versa). Specifically, we define identity function $1_{\mathbb{R}G}$: $\mathbb{R} \rightarrow$ GF, defined as $1_{\mathbb{R}G}(x) = x^{\mathcal{F}}$, where the superscript $\mathcal{F}$ represents that the element is in a finite field. Similarly, the corresponding inverse identity function $1_{G\mathbb{R}}$: GF $\rightarrow \mathbb{R}$ can be defined as $1_{G\mathbb{R}}(x^{\mathcal{F}}) = x$.

A node k in RLNC can transmit

$$y(k) = \sum_{n=1}^{N} \bigoplus \{c_n(k) \otimes x_n^{\mathcal{F}}\}$$
$$\triangleq \{c_1(k) \otimes x_1^{\mathcal{F}}\} \oplus \{c_2(k) \otimes x_2^{\mathcal{F}}\} \oplus \cdots \oplus \{c_N(k) \otimes x_N^{\mathcal{F}}\}$$

which is a linear combination of $x_n^{\mathcal{F}}$ and coding coefficients $c_n(k)$. $\oplus$ and $\otimes$ denote an additive operation and a multiplicative operation defined in GF, respectively and $y(k)$ and $c_n(k)$ are always considered as an element in GF. The coding coefficients are uniformly and randomly chosen from GF with size 2^M, denoted by GF(2^M). The packets generated in each node are transmitted to its neighboring nodes towards the client nodes. We assume that the size of the input set is $|\mathcal{X}| \le 2^M$, which enables coding operations to be properly operated in GF(2^M).[1]

If K innovative (i.e., linearly independent) symbols or packets $y(1), \ldots, y(K)$ are available at a decoder, a linear system $\mathbf{y} = \mathbf{C} \odot \mathbf{x}$ can be formed as

$$[y(1), \ldots, y(K)]^T = \sum_{n=1}^{N} \bigoplus \{\mathbf{c}_n \otimes x_n^{\mathcal{F}}\}. \tag{1}$$

[1]If $|\mathcal{X}| > 2^r$, the input set is reduced (using e.g., source quantization), such that the resized input set does not exceed the predetermined GF size.

The $K \times N$ matrix $\mathbf{C}$ is referred to as the coding coefficient matrix, which consists of column vectors $\mathbf{c}_n = [c_n(1), c_n(2), \cdots, c_n(K)]^T$. If the coding coefficient matrix $\mathbf{C}$ is full-rank (i.e., $K = N$), then $\mathbf{x}$ is uniquely determined because the inverse of a coding coefficient matrix $\mathbf{C}^{-1}$ is also uniquely determined. The inverse of a coding coefficient matrix can be efficiently implemented based on well-known methods such as the Gaussian elimination over a GF. If the number of received symbols is insufficient to determine a unique $\mathbf{C}^{-1}$, i.e., $K < N$, approximate decoding approaches [7] can be deployed for reconstructing the source symbols, which is briefly discussed in the next section.

B. Overview of Approximate Decoding Algorithms for Correlated Sources

As discussed, if the number of received symbols is insufficient (i.e., $K < N$), the correlation coefficient matrix $\mathbf{C}$ is not full-rank. Hence, it is infeasible to find a unique $\mathbf{C}^{-1}$. The key idea of the approximate decoding algorithm is to make the correlation coefficient matrix $\mathbf{C}$ full-rank by using source correlation. In the approximate decoding algorithm, the correlation of the input data can be exploited to set additional constraints, or matrix $\mathbf{D}$, in the decoding process. The additional constraints in $\mathbf{D}$ are determined based on the correlation model between the input data. For example, in [7], $\mathbf{D}$ is constructed such that $x_i = x_j$ for the best matched data. More specifically, in $(N-K)\times N$ matrix $\mathbf{D}$, each row consists of zeros (i.e., additive identity of GF(2^M)) except two elements of value “1” and “1”, as “1” is an additive inverse of “1” in GF(2^M). This corresponds to the positions of the best matched data $x_i, x_j \in \mathcal{X}$. An approximation $\hat{\mathbf{x}}$ of the original data is correspondingly found as

$$\hat{\mathbf{x}} = 1_{G\mathbb{R}}\left(\hat{\mathbf{x}}^{\mathcal{F}}\right) = 1_{G\mathbb{R}}\left(\begin{bmatrix} \mathbf{C} \\ \mathbf{D} \end{bmatrix}^{-1} \odot \begin{bmatrix} \mathbf{y} \\ \mathbf{0}^{\mathcal{F}}_{(N-K)} \end{bmatrix}\right) \tag{2}$$

where $\mathbf{0}^{\mathcal{F}}_{(N-K)}$ is a vector with $(N-K)$ zeros.

While constructing matrix $\mathbf{D}$ in [7] is based on the best matched data, which may not correctly and accurately represent the actual source correlation, the approximate decoding algorithm in [8] attempts to consider more systematic model for constructing $\mathbf{D}$. The approach is based on a linear relationship, i.e.,

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta \cdot \mathbf{1} \tag{3}$$

where $\mathbf{1}$ and $\Delta \in \mathbb{Z}$ denote a vector with all ones and the similarity factor, respectively. As shown in (3), the similarity factor can be obtained by

$$\Delta \cdot \mathbf{1} = \mathbf{x}_{n+1} - \mathbf{x}_n$$

which is simply the difference between consecutive source data. This can represent how much two sets of data are similar and is in the range of $0 \le \Delta < 2^M$. Using the similarity factor Δ, the approximate decoding can be represented by

$$\hat{\mathbf{x}} = 1_{G\mathbb{R}}\left(\begin{bmatrix} \mathbf{C} \\ \mathbf{D} \end{bmatrix}^{-1} \odot \begin{bmatrix} \mathbf{y} \\ \Delta^{\mathcal{F}} \cdot \mathbf{1}_{(N-K)} \end{bmatrix}\right) \tag{4}$$

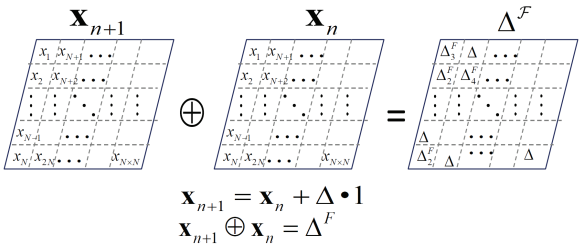

Fig. 1. An illustrative example of data structure in the framework. Each data set $\mathbf{x}_n$ is composed of $N \times N$ elements and each subscript of an element (e.g., '1' in x_1 or '2' in x_2) is referred to as a position index of data set $\mathbf{x}_n$.

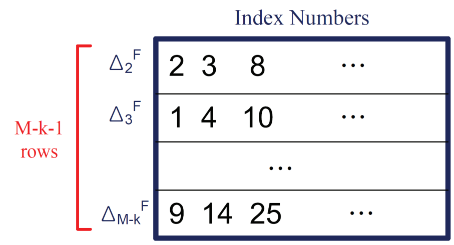

Fig. 2. An example of PIM structure. It consists of index numbers.

where $\mathbf{1}_{(N-K)}$ is an $(N-K)$ vector with all 1's and $\Delta^{\mathcal{F}} = 1_{\mathbb{R}G}(\Delta)$. Unlike the approximate decoding algorithm in (2), the approximate decoding algorithm shown in (4) explicitly considers the similarity of the source data.

Note that the information about Δ is sufficient for the perfect recovery of the original data in the field of real numbers, i.e., $\mathbf{x}_{n+1} - \mathbf{x}_n = \Delta \cdot \mathbf{1}$. However, the corresponding operation in GF, $\mathbf{x}_{n+1} \oplus \mathbf{x}_n$, may result in not only Δ but also the other different values[2], leading to a performance degradation. In order to improve the performance, the set of results computed from $\mathbf{x}_{n+1} \oplus \mathbf{x}_n$ is denoted as an $N \times N$ matrix $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$. Each element of this matrix is represented by $\Delta_n^{\mathcal{F}}$, where n is a position index. Therefore, there can be n different elements, i.e., $\Delta_1^{\mathcal{F}}, \Delta_2^{\mathcal{F}}, \ldots, \Delta_n^{\mathcal{F}}$, in $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$. Note that $\Delta_1^{\mathcal{F}} = \Delta^{\mathcal{F}}$ and $\Delta^{\mathcal{F}} = 1_{R\mathbb{G}}(\Delta)$. For example, while a single value 3 in the field of real number (i.e., $x_{n+1} - x_n = 3 = \Delta$) may induce a set of candidates such as 3, 5, 7, 13, 15, 29 and 31 in GF(2^5) by the operation of $x_{n+1} \oplus x_n$. Fig. 1 shows an illustrative example of the above data structure.

Since this characteristics has not been considered in the prior implementations of the approximated decoding (e.g., [7], [8]), the performance was limited. In order to improve the performance, we explicitly consider this characteristics and propose to use it as a form of PIM in the implementation of the approximate decoding.

III. Approximate Decoding with Position Information Matrix (PIM)

In this section, we propose a structure of PIM and discuss how to deploy the PIM in the framework of the approximate decoding. We then show that PIM can lead to an improved performance. Recall that we consider the source data correlation, which is modeled as a linear correlation, i.e., $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta \cdot \mathbf{1}$.

A. Implementation of Approximate Decoding using PIM

As discussed, there can be several different values of $\Delta_n^{\mathcal{F}}$ in $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}} = \mathbf{x}_{n+1} \oplus \mathbf{x}_n$. Thus, the information about different $\Delta_n^{\mathcal{F}}$ and the corresponding positions indices in the data set, $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$, can be transmitted to the decoder for better approximate decoding. The information is delivered as a form of PIM, which includes indices of positions for potential candidates in $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$. An illustrative PIM structure is shown in Fig. 2.

Based on the PIM, the prior approximate decoding algorithm shown in (4) can be improved by explicitly utilizing the information included in the PIM. Specifically, Δ in (4) is replaced with $\boldsymbol{\Delta}_{\boldsymbol{PIM}}^{\boldsymbol{\mathcal{F}}}$, which is $(N-K)$ vector with PIM, i.e.,

$$\hat{\mathbf{x}} = 1_{G\mathbb{R}} \left(\begin{bmatrix} \mathbf{C} \\ \mathbf{D} \end{bmatrix}^{-1} \odot \begin{bmatrix} \mathbf{y} \\ \boldsymbol{\Delta}_{\boldsymbol{PIM}}^{\boldsymbol{\mathcal{F}}} \end{bmatrix} \right). \tag{5}$$

In order to implement the algorithm in (5), the vector $\boldsymbol{\Delta}_{\boldsymbol{PIM}}^{\boldsymbol{\mathcal{F}}}$ is initialized by $\Delta^{\mathcal{F}} \cdot \mathbf{1}_{(N-K)}$. Then, every element in $\boldsymbol{\Delta}_{\boldsymbol{PIM}}^{\boldsymbol{\mathcal{F}}}$ corresponding to PIM is replaced by $\Delta_n^{\mathcal{F}}$. Note that the index numbers in nth row of PIM mean $\Delta_{n+1}^{\mathcal{F}}$ (see Fig. 2 for example), which are computed by Property 1.

The PIM can thus provide more refined information, leading to performance improvement of the approximate decoding. We next study several properties of the approximate decoding with PIM.

B. Properties of Approximate Decoding with PIM

In [8], it is shown that the performance of the approximate decoding is maximized when similarity factor Δ has a form of 2^k. Thus, we first start to study the case $\Delta^{\mathcal{F}} = 2^k$. In the following proofs, $B_{x(k)}$ and $\overline{B}_{x(k)}$ represent 1 and 0 at the kth bit position from LSB of x, respectively. The rest of bits do not matter with this analysis. In addition, δ is defined as a result of $x_n \oplus x_{n+1}$, i.e., $\delta = x_n \oplus x_{n+1}$, which represents elements in $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$.

Property 1. *If $\Delta^{\mathcal{F}} = 2^k$ $(0 \leq k < M)$ in GF(2^M), there are at most $(M-k)$ candidates of $\boldsymbol{\Delta}^{\mathcal{F}}$, where nth candidate $\Delta_n^{\mathcal{F}}$ is expressed as*

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

and its probability is given by

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

Proof: An element in GF(2^M) can be expressed as M bits. When $\Delta^{\mathcal{F}} = 2^k$, only one bit at $(k+1)$th position is set by 1, i.e., $B_{x(k+1)}$, and the rest of bit positions are set

[2]The addition and subtraction in GF are the equivalent operations and they are performed by XOR (exclusive OR) in this paper.

by 0. Since zeros do not influence the XOR operation (recall that zeros are additive identity in GF), we need to consider the positions where the bits are set by '1'. A candidate of $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}(\neq \Delta^{\mathcal{F}})$ may be generated if 1 is set for both $(k+1)$th bit position of $\Delta^{\mathcal{F}}$ and source data x, i.e., $B_{\Delta^{\mathcal{F}}(k+1)}$ and $B_{x(k+1)}$. Since carriage return from $(k+1)$th bit to $(k+2)$th bit is generated, $\Delta_2^{\mathcal{F}}$ is generated, which is expressed as $2^{k+1}+2^k$. If $B_{x(k+2)}$, another candidate of $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$ denoted by $\Delta_3^{\mathcal{F}}$ is generated and $\Delta_3^{\mathcal{F}} = 2^{k+2}+2^{k+1}+2^k$. This process continues until $\overline{B}_{x(M)}B_{x(M-1)}\cdots B_{x(k+1)}$. Therefore, there can be at most $M-k$ candidates of $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$ and nth candidate is expressed as $\Delta_n^{\mathcal{F}} = \sum_{i=k}^{k+n-1} 2^{k+i}$.

We then find the probability mass function for $\Delta_n^{\mathcal{F}}$. If $\overline{B}_{x(k+1)}$, $\Delta_1^{\mathcal{F}}$(=$\Delta^{\mathcal{F}}$= 2^k) is generated with probability of

$$\Pr(\delta = \Delta_1^{\mathcal{F}}) = \frac{2^{M-1}}{2^M - 2^k}.$$

This is because there are at most $2^M - 2^k$ cases in total while there are 2^{M-1} cases when $\overline{B}_{x(k+1)}$. By generalizing this, if $\overline{B}_{x(k+n)}B_{x(k+n-1)}\cdots B_{x(k+1)}$, for $2 \le n \le M-k$, $\Delta_n^{\mathcal{F}}$ is generated with probability of

$$\Pr(\delta = \Delta_n^{\mathcal{F}}) = \frac{2^{M-n}}{2^M - 2^k}.$$

Therefore, $\Pr(\delta = \Delta_n^{\mathcal{F}})$ for $1 \le n \le M-k$ is expressed as

$$\Pr(\delta = \Delta_n^{\mathcal{F}}) = \frac{2^{(M-n)}}{2^M - 2^k} \quad \text{for} \quad 1 \le n \le M-k. \tag{6}$$

■

The candidates of $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$ and the corresponding probabilities are shown in Table I when $\delta = 2^k$ in GF(2^M).

Based on Property 1, which provides a probability mass function for potential candidates, we can implement a PIM, where each row of a PIM represents each candidates. For example, the first row has the index numbers of $\Delta_2^{\mathcal{F}}$, the second row has the index numbers of $\Delta_3^{\mathcal{F}}$, and the $(M-k-1)$th row has the index numbers of $\Delta_{M-k}^{\mathcal{F}}$. Since it is known that $\Delta^{\mathcal{F}}$ has the highest probability among possible $\Delta_n^{\mathcal{F}}$, we may include only $(M-k-1)$ candidates except $\Delta^{\mathcal{F}}$ in PIM.

In this section, we show that PIM can provide more refined information about the correlation structure of the source data to the decoder, so that the performance of approximate decoding is improved. However, transmitting more information (i.e., more information is included in PIM) requires more communication overheads. Hence, it is essential to investigate the tradeoff between amount of information included in PIM and performance improvement. This is studied in Property 2.

Property 2. *As $(n-1)$th row of PIM (i.e., $\Delta_n^{\mathcal{F}}$) is additionally used in the approximate decoding, its performance improves by $1/2^{(n-1)}$.*

Proof: As shown in Table I, the probability of $\delta = \Delta_n^{\mathcal{F}}$ is given by $\frac{2^{(M-k-n)}}{2^{(M-k)}-1}$. This means that the probability decreases by $1/2^n$ as n increases. Thus, $\Delta_n^{\mathcal{F}}$ is additionally used for approximate decoding, the performance can be improved by $1/2^{(n-1)}$ times of the last performance improvement. ■

Note that the performance improvement by using additional information in PIM is a result of "additional correction".

We thus far have studied several properties when $\Delta^{\mathcal{F}} = 2^k$. If $\Delta^{\mathcal{F}} \neq 2^k$, there are a large number of different cases, and hence, we provide an algorithm to find candidates of $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$. This is presented in Algorithm 1.

Algorithm 1 Finding candidates of $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}$

Given: received similarity factor Δ, GF size 2^M.
if $\Delta = 2^k$, **then**
  use Property 1
else // $\Delta \neq 2^k$
  Find $p_1, p_2, p_3, \ldots, p_n$
  such that $\Delta = 2^{p_1} + 2^{p_2} + 2^{p_3} + ... + 2^{p_n}$,
  for $M > p_1 > p_2 > p_3 > \ldots > p_n \geqq 0$
  Find all candidates $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}(\mathbf{2}^{\boldsymbol{p_n}})$ for each 2^{p_n} based on Property 1.
  Find Cartesian product set S,
  $S = \boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}(\mathbf{2}^{\boldsymbol{p_1}}) \times \boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}(\mathbf{2}^{\boldsymbol{p_2}}) \times \ldots \times \boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}(\mathbf{2}^{\boldsymbol{p_n}}) = \{(\gamma_1, \ldots, \gamma_n), \quad \gamma_i \in \boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}}(\mathbf{2}^{\boldsymbol{p_i}})\}$
if γ_n exists even times in S, **then**
  Eliminate γ_n in S
else // odd times
  Choose only one γ_n in S
$\boldsymbol{\Delta}^{\boldsymbol{\mathcal{F}}} \leftarrow S$
end if

IV. Simulation Results

In this section, we implement approximate decoding in conjunction with PIM and show that using PIM can achieve an improved performance. In our simulation, we consider three correlated source data sets. The data sets are generated such that they are linearly correlated with the similarity factor of Δ and the experiments are performed independently 1000 times. We set Δ as 8 and GF size is 2^{10}. We generate 10 packets by default and change the packet loss rate.

Fig. 3 shows the performance of approximate decoding algorithm using the PIM. The results present normalized average error rates and thus lower values imply smaller distortion (or better performance). It is clear that the proposed approximate decoding with PIM outperforms the algorithm presented in [8] (i.e., PIM is not deployed) over all the ranges of packet loss rates. However, in both algorithms, it is observed that the performance of approximate decoding significantly degraded as the packet loss rates increases more than 50%. This is because the approximate decoding algorithm has to use more "artificially" added packets than the originally network coded data. It is also observed that there are several regions of packet loss rates where the performance of the approximate decoding algorithm is saturated. In particular, this happens when PIM has less than $3\Delta_n^{\mathcal{F}}$s. Thus, PIM can be adaptively structured by explicitly considering the packet loss rates and a target performance.

Property 2 is also confirmed in Fig. 4. In this experiment,

	$\Delta_1^{\mathcal{F}}$	$\Delta_2^{\mathcal{F}}$	$\cdots$	$\Delta_n^{\mathcal{F}}$	$\cdots$	$\Delta_{M-k}^{\mathcal{F}}$
$\boldsymbol{\Delta}^{\mathcal{F}}$	2^k	$2^{k+1}+2^k$	$\cdots$	$2^{k+(n-1)}+2^{k+(n-2)}+...+2^{k+1}+2^k$	$\cdots$	$2^{M-1}+...+2^{k+1}+2^k$
Probability	$\frac{2^{(M-1)}}{2^M-2^k}$	$\frac{2^{(M-2)}}{2^M-2^k}$	$\cdots$	$\frac{2^{(M-n)}}{2^M-2^k}$	$\cdots$	$\frac{2^k}{2^M-2^k}$

TABLE I
CANDIDATES OF $\boldsymbol{\Delta}^{\mathcal{F}}$ AND PROBABILITIES TO BE OBSERVED WHEN Δ= 2^k IN GF(2^M).

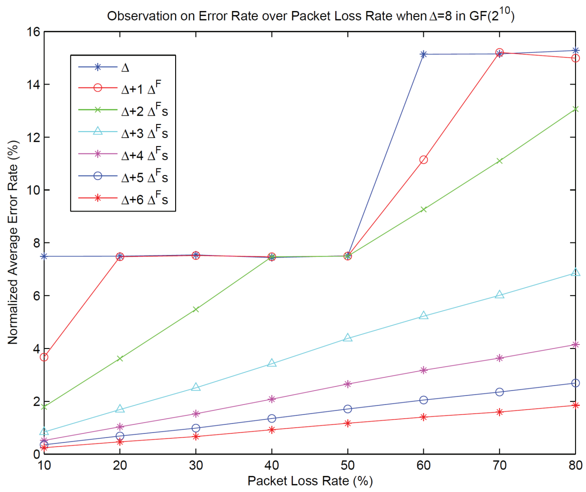

Fig. 3. Measured normalized average error rate of the system without and with PIM. $\Delta = 8$ and $GF(2^{10})$ are used in this simulation. At most six $\boldsymbol{\Delta}^{\mathcal{F}}$(PIM) can be provided in decoder through PIM.

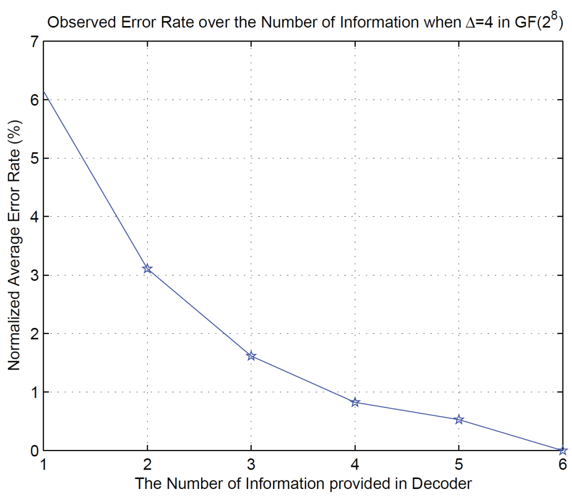

Fig. 4. Normalized average error rate as increasing the number of additional information when Δ=4.

we set Δ=4 in GF(2^8) with 1/3 packet loss rate. As discussed in Property 2, we can observe that the normalized average error rates decrease as the number of additional information increases. More specifically, there is 50% performance improvement (in terms of the normalized average error rates) in every additional information in PIM is deployed. We finally note that there is no error when six information in PIM is used. This is because that 6 information is the maximum number of information that can be included in PIM, as $M-k = 8-2 = 6$ (discussed in Property 1).

V. CONCLUSIONS

In this paper, we consider the deliver of correlated source data that are encoded by network coding technique with time constraints. When the number of received data packets is not enough for perfect data recovery, the approximate decoding algorithm can be deployed and. In this paper, we propose to use PIM as additional information to take into account the unique characteristics of the finite field operations, which improves the performance of the approximate decoding. We show that using PIM can lead to a better performance of the approximate decoding and analytically study the impact of PIM on the decoding performance. As a future research direction, besides the linear correlation in the source data, more generalized correlation models can be incorporated into the approximate decoding algorithm, such that it can be used for a variety of correlated source data.

REFERENCES

[1] R. R. Rout, S. K. Ghosh, and S. Chakrabarti, "A network coding based probabilistic routing scheme for wireless sensor network," in *Wireless Communication and Sensor Networks (WCSN)*, Dec. 2010.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *MobiCom'00*, Boston, MA, USA, Aug. 2000, pp. 56–67.

[3] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, Apr. 2002.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[5] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[6] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proc. Allerton Annual Conf. Commun., Control, and Comput.*, Monticello, IL, USA, Oct. 2003.

[7] H. Park, N. Thomos, and P. Frossard, "Transmission of correlated information sources with network coding," in *Proc. European Signal Processing Conference (EUSIPCO-2010)*, Aug. 2010, pp. 1389–1393.

[8] M. Kwon and H. Park, "An improved approximate decoding with correlated sources," in *SPIE 2011*, San Diego, CA, USA, Aug. 2011.