The Impact of Network Coding Cluster Size on Approximate Decoding Performance

Minhae Kwon[1] and Hyunggon Park[1]
[1] Department of Electronics Engineering, Ewha Womans University
Seoul – Republic of Korea
[e-mail: minhae.kwon@ewhain.net; hyunggon.park@ewha.ac.kr]
*Corresponding author: Hyunggon Park

Received July 28, 2015; revised October 16, 2015; revised December 17, 2015; revised January 17, 2015; accepted January 25, 2016; published March 31, 2016

Abstract

In this paper, delay-constrained data transmission is considered over error-prone networks. Network coding is deployed for efficient information exchange, and an approximate decoding approach is deployed to overcome potential all-or-nothing problems. Our focus is on determining the cluster size and its impact on approximate decoding performance. Decoding performance is quantified, and we show that performance is determined only by the number of packets. Moreover, the fundamental tradeoff between approximate decoding performance and data transfer rate improvement is analyzed; as the cluster size increases, the data transfer rate improves and decoding performance is degraded. This tradeoff can lead to an optimal cluster size of network coding-based networks that achieves the target decoding performance of applications. A set of experiment results confirms the analysis.

Keywords: Network coding, cluster size, network optimization, approximate decoding, Internet of Things (IoT), wireless sensor networks (WSN)

This research was supported in part by the Ministry of Science, ICT and Future Planning (MSIP), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2015-H8501-15-1007) supervised by the Institute for Information & Communications Technology Promotion (IITP) and in part by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP; No. NRF-2014R1A2A1A11051257).

http://dx.doi.org/10.3837/tiis.2016.03.011 **ISSN : 1976-7277**

1. Introduction

The new era of communication and computer networks can be represented by always-connected devices such as the Internet of Things (IoT), i.e., everyday objects are connected to a network so that data can be shared among them. Supported by the hardware development of sensors and communication chipsets, many devices have become communication enabled [1]–[5]. This has resulted in explosive data generation; hence, IoT networks should be able to efficiently manage a large amount of data, i.e., efficient information exchange and delivery in ad hoc network topologies.

Network coding can be used as a solution to enable efficient information exchange and delivery [6]. Such coding can increase the data transfer rate by utilizing path diversity in networks. Instead of simply forwarding data as in conventional routing, network coding enables intermediate nodes to combine incoming data packets into a single packet based on basic operations and to forward the packet to neighbor nodes [7]–[9]. The potential advantages of network coding include efficiency in resources (e.g., bandwidth and power), robustness against network dynamics [10], and scalability [11]. However, network coding has a critical drawback when deployed in delay-constrained error-prone networks (e.g., disaster/emergency networks). Since multiple-source data sets are combined in a network-coded packet, decoding is permitted only when receiving a sufficient number of encoded packets (i.e., at least the same as the number of combined source data sets). If there are not enough packets for decoding, none of the source data sets can be recovered. This is referred to as the *all-or-nothing* nature of network coding [12]. In order to overcome this limitation, approximate decoding has been proposed [13]–[16]. Approximate decoding enables the source data to be recovered even when the number of received packets is not sufficient at the moment of reconstruction.

An important issue to be resolved is the efficient formation of clusters when network coding is deployed in error-prone networks [17]–[24]. Most clustering studies have focused on cluster formation and cluster head selection, which can lead to minimum energy consumption, and there are few studies on determination of *cluster size*, particularly when network coding is deployed. This is a fundamental question because of the network coding operations that combine data packets collected from the cluster members in each cluster. Therefore, the number of cluster members (i.e., cluster size) should be taken into account in cluster formation while explicitly considering the delay constraints of the application and decoding performance.

It is intuitively expected that a larger cluster size will lead to better efficiency in terms of data transfer rate as more source data packets are combined and transmitted together. However, as cluster size increases, decoders may need to wait longer to receive enough packets to decode, which incurs longer decoding latency. Moreover, the approximate decoding performance is determined by the cluster size because any packets missed during the transmission can significantly reduce the number of correctly recovered source data sets encoded together. Therefore, it is essential to analytically investigate the impact of cluster size on approximate decoding performance so that an optimal size of a cluster can be determined.

In this paper, the impact of cluster size on approximate decoding performance and data transfer rate is analytically studied. In particular, the case in which packets are lost or delayed by the decoding deadline is mainly considered, a situation which is highly probable for delay-constrained data transmission over error-prone networks. An analytical trade-off between approximate decoding performance and data transfer rate is shown, i.e., a smaller

cluster size achieves better performance but demonstrates degraded data transfer rate improvement.

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

- we quantify the approximate decoding performance,
- we show that the performance is determined only by the number of packets,
- we analytically study the impact of cluster size on approximate decoding performance and data transfer rate, and
- we show the tradeoff between approximate decoding performance and data transfer rate.

This paper is organized as follows. In Section 2, related works are discussed. The system setup and a brief overview of approximate decoding are provided in Section 3. The performance analysis of approximate decoding and the impact of cluster size on decoding performance are studied in Section 4.1 and Section 4.2, respectively. In Section 5, simulation results are presented. Finally, the conclusion is drawn in Section 6.

2. Related Works

In this section, prior works related to the proposed approaches are presented. In order to overcome the all-or-nothing problem of network coding, several approaches have been studied. In-network compression has been developed in several studies [25]–[28]. Motivated by the compressed sensing theory, the number of packets to be transmitted can be decreased via compression processes in networks, and a decoder reconstructs original data from the compressed packets. In [25], correlated sources are considered for utilizing compressed sensing in source and channel coding processes. In [26], encoders combine source data based on compressive measurements, and statistical dependency is used with the sum-product algorithm for reconstruction. A practical system for exploiting source correlation knowledge is provided in [27], and an approach to combine the field difference between network coding and compressed sensing, which are a Galois Field (GF) and real field, respectively, is presented in [28]. In these works, however, it is still possible that compressed packets are not delivered to the decoder on time, leading to decoding failure, even though the number of packets used for the decoding process is less than the number of original packets.

As an alternative approach for overcoming the all-or-nothing problem, approximate decoding has been developed [13]–[16]. Approximate decoding was originally proposed in [13] with a heuristic approach. The source data similarity is used at the decoder, and the optimal size of the finite coding field is determined. In [14], a linearly correlated source and corresponding decoder design are provided, and the impact of the similarity factor is analyzed. In order to improve the decoding performance of approximate decoding, a position information matrix (PIM) is used [15]. The PIM allows decoders to refine the recovered data and to improve decoding performance. If the distribution of the source correlation is symmetric, the knowledge of the mean of distribution is sufficient to maximize approximate decoding performance [16]. Even though these works provide solutions to the all-or-nothing problem, they do not consider cluster formations in networks, which is essential for efficiently managing IoT networks. Cluster formation should be studied by explicitly considering several parameters such as cluster size because they might significantly affect network coding and decoding performance.

For efficient cluster formation in error-prone networks, several algorithms have been developed while minimizing energy consumption in the networks. Low-Energy Adaptive Clustering Hierarchy (LEACH) [22] was one of the first hierarchical routing approaches. In this algorithm, cluster heads are randomly selected, so the performance of the algorithm greatly relies on cluster heads rather than cluster members. In order to efficiently select cluster heads, Low-Energy Adaptive Clustering Hierarchy Centralized (LEACHC) is presented in [23] to use information about locations and energy levels of nodes that belong to base stations for cluster formation. Hybrid Energy-Efficient Distributed clustering (HEED) [24] was proposed with use of a multihop clustering algorithm, which determines cluster heads based on the residual energy of each node and the intra-cluster communication cost. However, none of the algorithms mentioned above consider deploying network coding techniques in error-prone

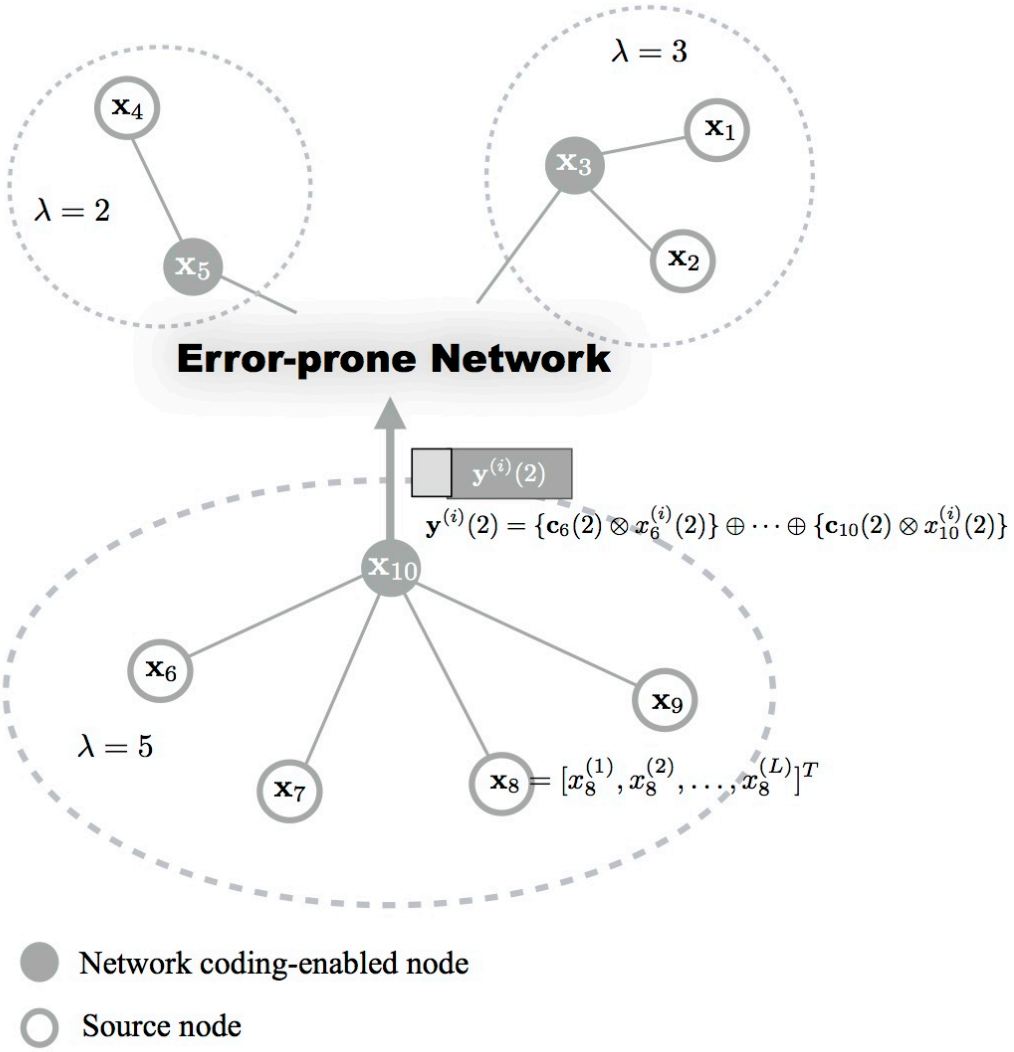

Fig. 1. An illustrative example of an error-prone network based on network coding. In this example, three clusters involve 10 source nodes. A network coding-enabled node collects data from its cluster members and performs network coding.

networks. Therefore, a blind deployment of these algorithms to network coding-based data delivery may provide only limited performance.

3. System Setup

An error-prone network consists of source nodes, intermediate nodes, and a destination. The nodes form clusters and perform network-coding operations. The network-coded data are delivered to the destination through intermediate nodes that also perform network-coding operations. Our analysis is based on a single cluster, which can be extended to multiple clusters. Parts of the system setup discussed in this section can also be found in [13] and [15]. An illustrative example of the considered error-prone network is shown in **Fig. 1**.

3.1 Linearly Correlated Sources

Let $\mathbf{x}_t = [x_t^{(1)}, \ldots, x_t^{(i)}, \ldots, x_t^{(L)}]^T$ be the t-th source data set obtained by the t-th source node and its element, $x_t^{(i)}$, for $1 \le i \le L$ be the i-th element in $\mathbf{x}_t$. All source data are in $GF(2^M)$ [1], which is a GF with a size of 2^M, such that network-coding operations can be performed in $GF(2^M)$. In this paper, source data sets are linearly correlated [29], [30], i.e.,

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta_1 \cdot \mathbf{1} \tag{1}$$

where $\mathbf{1}$ denotes a vector with all ones, and $\Delta_1 = 2^k$, $0 \le k < M$, represents the source correlation. This source model can capture several types of signal such as temperature changes in long-term periods and seismic signals at different sources.

In the field of real numbers ($\mathbb{R}$), Δ_1 can perfectly capture the relationship between $\mathbf{x}_{t+1}$ and $\mathbf{x}_t$, as $\mathbf{x}_{t+1} - \mathbf{x}_t = \Delta_1 \cdot \mathbf{1}$ is deterministic. However, results of the corresponding operation in the GF, $\mathbf{x}_{t+1} \oplus \mathbf{x}_t$ [2], can be determined in a set, $\mathbf{\Delta}_t$, expressed as

$$\begin{aligned} \mathbf{\Delta}_t &= \{\mathbf{x}_{t+1} \oplus \mathbf{x}_t \mid \mathbf{x}_{t+1} - \mathbf{x}_t = \Delta_1 \cdot \mathbf{1}\} \\ &\ni \Delta_1, \Delta_2, \ldots, \Delta_i, \ldots, \Delta_n \end{aligned} \tag{2}$$

where $n = M - k$ for $\Delta_1 = 2^k$ [15]. Therefore, unlike the case in $\mathbb{R}$, the correlation between consecutive source data sets can be captured by considering $\mathbf{\Delta}_t$. This problem has been addressed in [15], and a PIM is introduced as including elements in $\mathbf{\Delta}_t$ and their positions. The PIM is constructed at a source data set and transmitted to the decoder along with data packets.

---

[1] As in [13], an identity function is defined for the field transition between $\mathbb{R}$ and $GF(2^M)$. If an obtained source symbol is not in $GF(2^M)$, the identity function can be used prior to our system setup.

[2] Addition and subtraction are denoted by $\oplus$ and are equivalent operations in the GF. In this paper, they are performed by XOR (exclusive OR).

3.2 RLNC-based Encoding

An intermediate node at the h-th coding stage receives packets $\mathbf{y}^{(i)}(h-1)$ from other nodes and generates packets $\mathbf{y}^{(i)}(h)$ by mixing them based on random linear network coding (RLNC) [31]. Then, the node again transmits $\mathbf{y}^{(i)}(h)$ to its neighbor nodes toward the destination. Specifically, a set of K innovative (i.e., linearly independent) packets, denoted as $\mathbf{y}^{(i)}(h) = \left[y_1^{(i)}(h), \ldots, y_K^{(i)}(h) \right]^T$, is generated by

$$\begin{aligned} \mathbf{y}^{(i)}(h+1) &= \mathbf{c}(h) \odot \mathbf{y}^{(i)}(h) \\ &= \{\mathbf{c}_1(h) \otimes y_1^{(i)}(h)\} \oplus \cdots \oplus \{\mathbf{c}_\lambda(h) \otimes y_\lambda^{(i)}(h)\} \end{aligned} \tag{3}$$

which is a linear combination of $\mathbf{y}^{(i)}(h)$ and the coding coefficient matrix $\mathbf{c}(h) = [\mathbf{c}_1(h), \cdots, \mathbf{c}_\lambda(h)]^T$. λ is the number of packets combined together, which is the same as the number of members in a cluster, i.e., *cluster size*. The number of outgoing packets, K, is chosen such that $K \geq \lambda$ and may depend on the expected packet erasure rate; higher K is recommended for high erasure rate, and vice versa. Note that $\mathbf{y}^{(i)}(1) = [x_1^{(i)}, \ldots, x_\lambda^{(i)}]^T$ is the initial packet. $\odot$ denotes the multiplication between matrices in the GF, and $\oplus$ and $\otimes$ denote additive and multiplicative operations defined in the GF, respectively. In RLNC, the elements of $\mathbf{c}(h)$ are uniformly and randomly chosen from $GF(2^M)$.

Finally, the coded packet at the h-th coding stage in (3) can be expressed as

$$\begin{aligned} \mathbf{y}^{(i)}(h+1) &= \mathbf{c}(h) \odot \mathbf{y}^{(i)}(h) \\ &= \mathbf{c}(h) \odot \mathbf{c}(h-1) \odot \cdots \odot \mathbf{c}(1) \odot \mathbf{x}^{(i)} \\ &= \mathbf{C}(h) \odot \mathbf{x}^{(i)} \end{aligned} \tag{4}$$

where $\mathbf{C}(h)$ is referred to as a global coding coefficient matrix, which is included in the header of the packet and delivered to the decoder to enable decoding and reconstruction. As shown in [31], $\mathbf{C}(h)$ can be assumed to be full-rank when the GF size is larger than the number of receivers in RLNC networks. Hence, we assume that $\mathbf{C}(h)$ is full-rank in this paper.

3.3 Approximate Decoding with PIM

For a decoder at the destination (h_D-th coding stage), if the coding coefficient matrix, $\mathbf{C}(h_D - 1)$, is full-rank (i.e., $K = \lambda$), then $\hat{\mathbf{x}}^{(i)} = [\hat{x}_1^{(i)}, \ldots, \hat{x}_\lambda^{(i)}]^T$ can be uniquely determined as

$$\hat{\mathbf{x}}^{(i)} = \left[\hat{x}_1^{(i)}, \ldots, \hat{x}_\lambda^{(i)} \right]^T = \mathbf{C}(h_D - 1)^{-1} \odot \mathbf{y}^{(i)}(h_D). \tag{5}$$

However, if the number of received packets is insufficient to determine a unique $\mathbf{C}(h_D - 1)^{-1}$ (i.e., $K < \lambda$) as a result of packet delay and/or packet loss in transmission, for example, $\mathbf{C}(h_D - 1)$ is not full-rank, potentially leading to multiple solutions, $\hat{\mathbf{x}}^{(i)}$, to the linear system

expressed in (5). This problem was solved based on approximate decoding with the PIM [15], expressed as

$$\hat{\mathbf{x}}^{(i)} = \begin{bmatrix} \mathbf{C}(h_D - 1) \\ \mathbf{D} \end{bmatrix}^{-1} \odot \begin{bmatrix} \mathbf{y}^{(i)}(h_D) \\ \mathbf{\Delta}_{PIM} \end{bmatrix}. \tag{6}$$

The main idea of the approximate decoding algorithm is to add extra equations $\mathbf{D}$ and $\mathbf{\Delta}_{PIM}$ based on the source correlation, so that the matrix $\left[\mathbf{C}(h_D - 1)^T \mathbf{D}^T\right]^T$ in (6) becomes invertible. Therefore, equation $\mathbf{\Delta}_t = \mathbf{x}_{t+1} \oplus \mathbf{x}_t$ is added to provide source characteristics in (6). In particular, $(\lambda - K) \times \lambda$ matrix $\mathbf{D}$ is constructed such that each row consists of zeros (i.e., additive identity of $GF(2^M)$) except for two elements of value "1" (because 1 is the additive inverse of 1 in $GF(2^M)$) that correspond to the positions of the linearly correlated data, $x_t^{(i)}$ and $x_{t+1}^{(i)}$ [13]. Then, $\mathbf{\Delta}_{PIM}$ with a size of $(\lambda - K)$ is accordingly determined using the PIM received from the encoder[3].

While it is shown that a PIM can improve the performance of the approximate decoding approaches, the impact of cluster size on the performance of the approximate decoding is not clearly quantified. This is discussed in Section 4.

4. Impact of Cluster Size on Approximate Decoding Performance

In this section, the impact of cluster size on data transfer rate and performance of the approximate decoding algorithm is studied in conjunction with the PIM.

4.1 Performance Analysis of Approximate Decoding

For the performance analysis, let $N_l := \lambda - K$ packets be unavailable at a decoder, i.e., the received packets are not sufficient for perfect decoding. Hence, the approximate decoding algorithm needs to be deployed. The performance of the approximate decoding algorithm is measured by the probability of data being correctly decoded, i.e., $\Pr(\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)})$. The main result is stated in the property shown below.

Property: The probability that data is correctly decoded based on the approximate decoding with the PIM depends only on N_l. Furthermore, the performance improves as N_l decreases.

Proof: See Appendix A.

[3] In order to avoid duplication of the description from prior works, we present only the key idea of a PIM in this paper. More information can be found in [16].

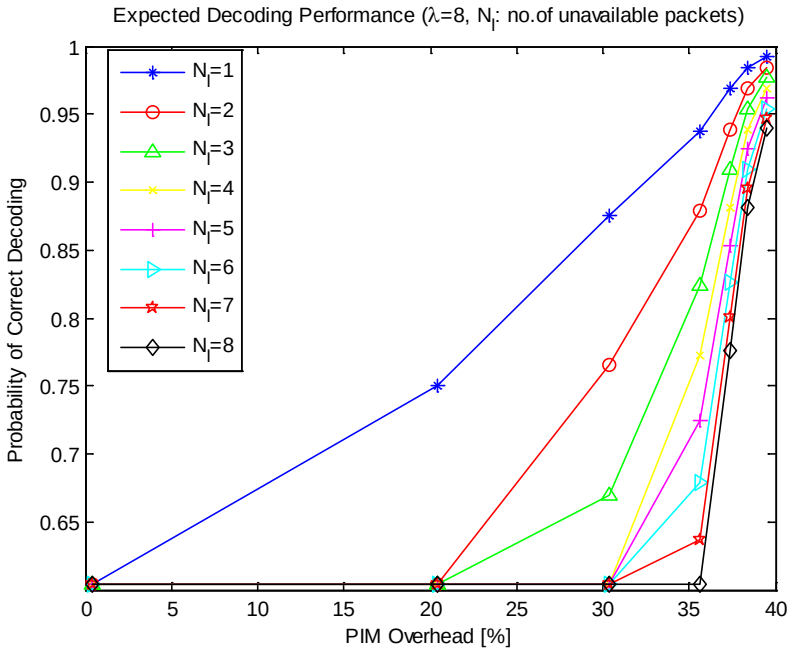

Fig. 2. As N_l decreases, the proposed performance measure (probability of correct decoding) increases over various PIM overhead ranges.

An illustrative example that confirms the property for various PIM overheads is shown in **Fig. 2**. The PIM overhead represents the ratio between the amount of information additionally included in a PIM and the amount of data needed to be transmitted. The probability of correct decoding is computed based on (15) in Appendix A. **Fig. 2** shows that smaller N_l leads to higher probability of correct decoding for all PIM overheads, meaning better performance. Since the performance of the approximate decoding with a PIM is bounded by a minimum performance level, θ [15], the plots shown in **Fig. 2** are generated by

$$\max\{\theta, \Pr(\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} \mid \Delta_1, \ldots, \Delta_n, N_l)\} \tag{7}$$

where $\lambda = 8$ and $\theta = 0.6042$.

We next consider the impact of cluster size on approximate decoding and network coding.

4.2 Impact of Cluster Size on Performance

In this section, the impact of cluster size λ on both approximate decoding performance and data transfer rate is investigated based on the property discussed in Section 4.1.

Given N_l, a packet loss rate of network condition γ is defined as

$$\gamma = N_l / \lambda. \tag{8}$$

Data transfer rate is defined as the amount of information that can be transmitted in a time slot, which is denoted by R and is expressed as

$$R = \frac{\lambda \cdot L \cdot M}{T_d} [\text{bits/sec}] \tag{9}$$

where T_d is the duration of the time slot. Since a source data is represented by M bits (as GF size is 2^M) and a packet consists of L source data, $M \times L$ indicates the bits per packet. In terms of packet loss rate and data transfer rate, the property can be interpreted as follows.

- **Interpretation 1:** Given packet loss rate γ, smaller cluster size λ leads to better performance.
- **Interpretation 2:** Larger cluster size λ leads to better data transfer rate.

As shown in (8), λ is proportional to N_l for fixed γ. Thus, a smaller λ can achieve better performance (Interpretation 1). Moreover, R is proportional to λ as in (9). Hence, the data transfer rate increases as λ increases (Interpretation 2).

The interpretations confirm a fundamental tradeoff between potential data transfer rate and performance of the approximate decoding, i.e., high data transfer rates can be achieved at the cost of decoding performance degradation, and vice versa. That is, a smaller cluster size leads to a higher probability of a sufficient number of packets being available for decoding, thereby achieving better approximate decoding performance. However, this does not take into account the advantages of deploying network coding techniques, i.e., data transfer rate improvement. Therefore, an appropriate cluster size is selected by taking into account the network conditions and the desired decoding performance.

5. Simulation Results

In this section, experimental results are presented and confirm the interpretations discussed in Section 4.2.

Fig. 3 shows the approximate decoding performance for several cluster sizes in error-prone networks with a 25% packet loss rate. In the simulations, parameters are set as $M = 10$, $L = 256$, $k = 3$, and $\gamma = 3$, meaning that the network-coding operations are performed in intermediate nodes based on RLNC in $GF(2^{10})$. The first set of source data, $\mathbf{x}_1$, with a data block size of 16×16, is randomly generated in the range of [0, $2^{10} - (\lambda - 1) \cdot \Delta_1 - 1$], and a set of linearly correlated source data is generated such that $\mathbf{x}_{t+1} = \mathbf{x}_1 + (t-1) \cdot \Delta_1 \cdot \mathbf{1}$, where $\Delta_1 = 8$. **Fig. 3** shows the average rates of correct decoding, defined as

$$\sum_{i=1}^{L} I(x_t^{(i)} - \hat{x}_t^{(i)}) / L \tag{10}$$

where

$$I(x) = \begin{cases} 1 & \text{if} \quad x = 0 \\ 0 & \text{if} \quad x \neq 0 \end{cases} \tag{11}$$

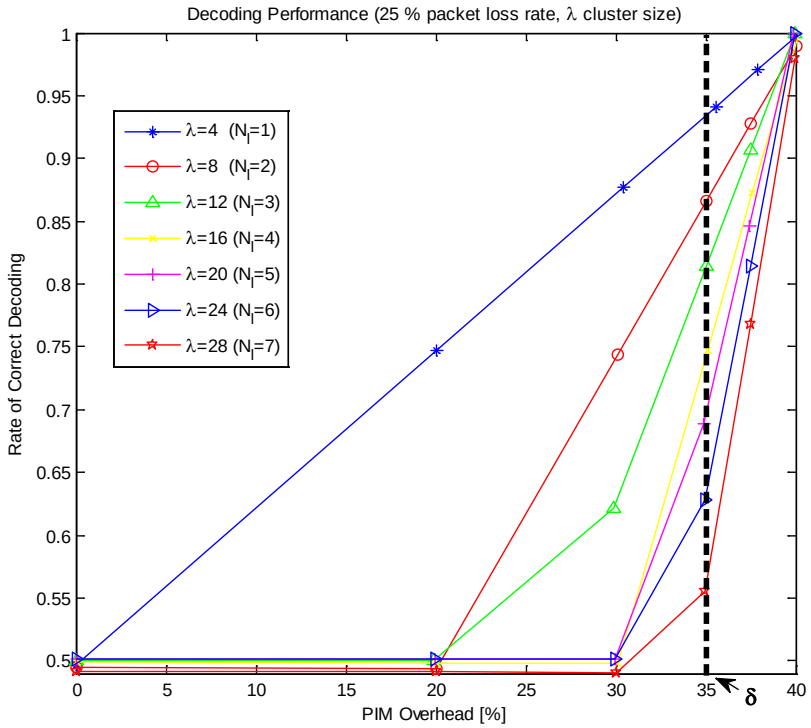

Fig. 3 The average rates of correct decoding for several PIM overheads given 25% packet loss rate ($\boldsymbol{\gamma = 0.25}$). As stated in Interpretation 1, smaller cluster size leads to better performances (i.e., higher correct decoding rates).

which indicates the ratio between the number of correctly decoded elements in $\mathbf{x}_t$ $(1 \le t \le T)$ and the total number of elements (L) in the source data sets over 1000 independent experiments.

Fig. 3 confirms the validity of Interpretation 1. Specifically, it is clear that smaller cluster size λ can generally lead to better performance. For example, if the PIM overhead is 35% (indicated by δ in **Fig. 3**), the best performance is achieved when $\lambda = 4$ (the smallest cluster size), while the performance is the worst when $\lambda = 28$ (the largest cluster size). Note that the plots for performances converge to similar levels in the ranges of very low PIM or very high PIM. This is because the information provided by the PIM is insufficient for approximate decoding to correctly recover the source data in the range of very low PIM. On the other hand, in the range of very high PIM, which corresponds to the case where $n = M - k$, all of the information needed by the approximate decoding algorithm for perfect decoding can be included in the PIM. Hence, the original source data symbols can be perfectly decoded.

Fig. 4 shows the fundamental tradeoff between cluster size and approximate decoding performance for several PIM overheads. In the simulations, parameters are set as $M = 10$, $L = 128$, $k = 3$, $n = 6$, and $T_d = 1$; thus, seven Δ_i ($i = 1, \cdots, 7$ as $M - k = 10 - 3 = 7$) can be included at most in a PIM [15]. The results shown in Fig. 4 include the cases where Δ_1, Δ_2, Δ_3, and Δ_4 are included in a PIM (corresponding to 30.6% PIM overhead) and the case where Δ_1, Δ_2, Δ_3, Δ_4, Δ_5, and Δ_6 (corresponding to 34% PIM overhead) are included in a PIM.

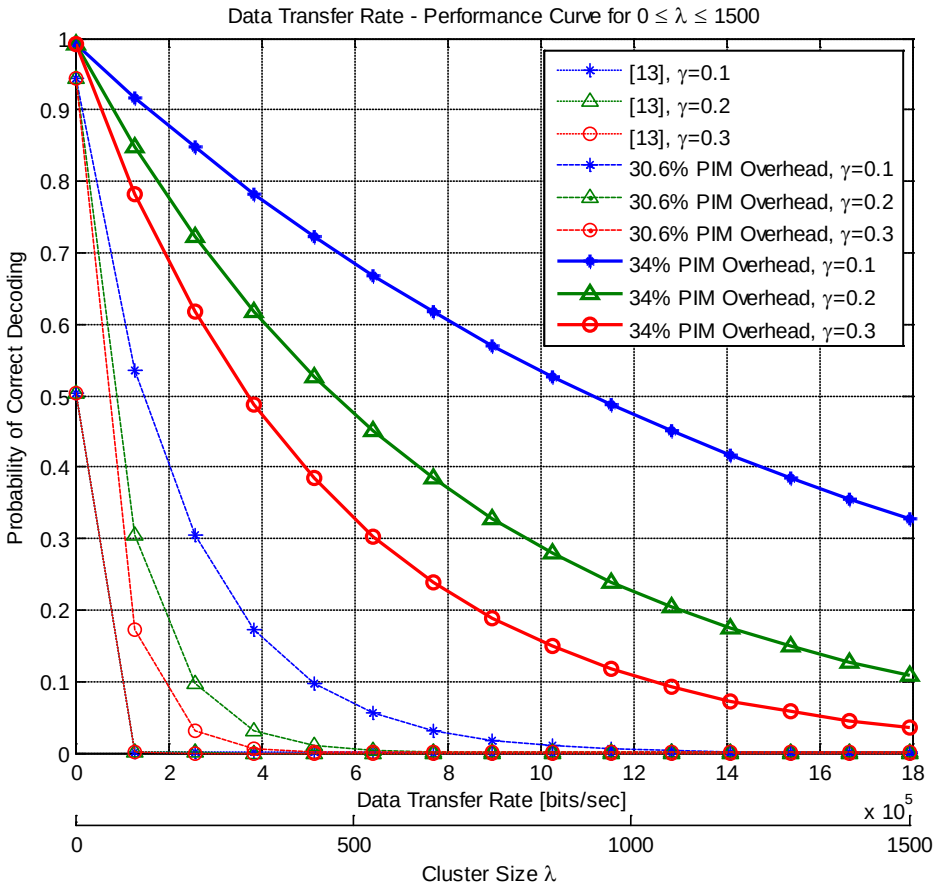

Fig. 4. Achieved results for trade-off between performance measure (probability of correct decoding) and data transfer rate according to (9).

The amount of PIM overhead can be computed as

$$\frac{\log_2 L}{M}\left(\sum_{i=2}^{n} 2^{-i}\right)\times 100 \ [\%] \tag{12}$$

if Δ_1, …, Δ_n are included in the PIM [15]. Based on (9), data transfer rate is linearly proportional to cluster size, i.e., $R = 128 \cdot 10 \cdot \lambda / 1$. Hence, the data transfer rates are presented together with cluster sizes in **Fig. 4**. The performance of the proposed approach is compared with that of an existing state-of-the art approach [13], which corresponds to the case of no PIM.

The simulation results indicate that the proposed approach always outperforms the existing algorithm [13], as the proposed approach is designed by considering the PIM and cluster size. More specifically, the probability of correct decoding significantly decreases as cluster size increases if packet loss occurs in transmission (i.e., $\gamma > 0$). If a PIM is provided, however, the probability of correct decoding improves as more PIMs are included. Moreover, it is observed that higher PIM overhead can lower the speed at which the probability of correct decoding degrades. Therefore, an optimal cluster size can be determined by taking into account the PIM overhead and a target decoding performance given network conditions (i.e., packet loss rates).

6. Conclusion

In this paper, the impact of cluster size on the approximate decoding performance and the data transfer rate is analytically investigated. The approximate decoding performance with a PIM is quantitatively evaluated, and it is shown that the performance only depends on the number of packets. Given the packet loss rates of networks, a smaller cluster size enhances the approximate decoding performance at the cost of data transfer rate degradation. Based on these findings, cluster sizes of error-prone networks can be optimized in order to meet target performance.

Appendix A

In Appendix A, the proof of the property in Section 4.1 is presented. Let Δ_R be a random variable for Δ_i. Recall that Δ_i is an element of $\mathbf{\Delta}_t = \mathbf{x}_{t+1} \oplus \mathbf{x}_t$, which describes the source correlation in the GF. We first consider the case where $\Delta = 2^k$. As shown in [15], the probability that $\Delta_R = \Delta_i$ is

$$\Pr(\Delta_R = \Delta_i) = \frac{2^{(M-k-i)}}{2^{(M-k)}-1} \tag{13}$$

where $1 \le i \le M-k$ and $GF(2^M)$. Thus, the probability that Δ_R is one of Δ_1, Δ_2, …, Δ_n $(1 \le n \le M-k)$ included in $\mathbf{\Delta}_t$ can be expressed as

$$\begin{aligned} \Pr(\Delta_R \in \{\Delta_1, \ldots, \Delta_n\}) &= \sum_{i=1}^{n} \Pr(\Delta_R = \Delta_i) \\ &= \sum_{i=1}^{n} \frac{2^{(M-k-i)}}{2^{(M-k)}-1}. \end{aligned} \tag{14}$$

When N_l packets are not available in the decoding process, the approximate decoding is deployed in conjunction with a PIM including the position information, $\Delta_2, \ldots, \Delta_n$. The probability of correct decoding when a PIM is provided can be expressed as

$$\begin{aligned} &\Pr(\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} \mid \Delta_1, \ldots, \Delta_n, N_l) \\ &= \prod_{j=1}^{N_l} \left(\sum_{i=1}^{n} \Pr(\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} \mid \Delta_i) \right) \\ &= \prod_{j=1}^{N_l} \left(\sum_{i=1}^{n} \Pr(\Delta_R = \Delta_i) \right) = \left(\sum_{i=1}^{n} \frac{2^{(M-k-i)}}{2^{(M-k)}-1} \right)^{N_l} \end{aligned} \tag{15}$$

In (15), the impact of Δ_i in a PIM on the decoding performance is assumed to be independent. Moreover, $\Pr(\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} \mid \Delta_i) = \Pr(\Delta_R = \Delta_i)$. Since n and M are given parameters by the encoder and k is determined by the source characteristics, the performance in (15) depends only on N_l. Furthermore, since $n \le M-k$ and $2^{M-k-i} / 2^{M-k} - 1$ is nonnegative,

$$\sum_{i=1}^{n} \frac{2^{M-k-i}}{2^{M-k}-1} \leq \sum_{i=1}^{M-k} \frac{2^{M-k-i}}{2^{M-k}-1} = \frac{1}{2^{M-k}-1} \sum_{i=1}^{M-k} 2^{M-k-i} = 1. \tag{16}$$

Based on (15), (16) can be written as

$$\Pr(\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} \mid \Delta_1, \ldots, \Delta_n, N_l) = \alpha^{N_l} \tag{17}$$

where $\alpha \leq 1$, concluding that the probability of correct decoding is a non-increasing function of N_l.

In the case where $\Delta \neq 2^k$, let $\Pr(\Delta_R = \Delta_i) = p_i$ in (13); correspondingly,

$$\Pr(\Delta_R \in \{\Delta_1, \ldots, \Delta_n\}) = \sum_{i=1}^{n} p_i \leq 1. \tag{18}$$

Then, $\Pr(\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} \mid \Delta_1, \ldots, \Delta_n, N_l)$ given in (15) can be expressed as

$$\Pr(\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} \mid \Delta_1, \ldots, \Delta_n, N_l) = \left(\sum_{i=1}^{n} p_i \right)^{N_l} \tag{19}$$

which also indicates that the probability of correct decoding is a non-increasing function of N_l.

Therefore, the performance of approximate decoding improves as N_l decreases, completing the proof.

■

References

[1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012. Article (CrossRef Link)

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013. Article (CrossRef Link)

[3] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "Iot gateway: Bridging wireless sensor networks into internet of things," in *Proc. of IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 347–352, 2010. Article (CrossRef Link)

[4] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer et al., "Internet of things strategic research roadmap," *Internet of Things-Global Technological and Societal Trends*, pp. 9–52, 2011.

[5] S. Hong, D. Kim, M. Ha, S. Bae, S. J. Park, W. Jung, and J.-E. Kim, "Snail: an ip-based wireless sensor network approach to the internet of things," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34–42, 2010. Article (CrossRef Link)

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000. Article (CrossRef Link)

[7] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003. Article (CrossRef Link)

[8] Z. Li, B. Li, D. Jiang, and L. C. Lau, "On achieving optimal throughput with network coding," in *Proc. of IEEE International Conference on Computer and Communications (INFOCOM)*, vol. 3, Miami, FL, USA, Mar, pp. 2184–2194, 2005. Article (CrossRef Link)
[9] P. A. Chou and Y. Wu, "Network coding for the internet and wireless networks," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 77–85, Sep. 2007. Article (CrossRef Link)
[10] T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. of IEEE International Symposium on Information Theory*, Cambridge, MA, USA, Jun/Jul. 2003. Article (CrossRef Link)
[11] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 829–848, Mar. 2006. Article (CrossRef Link)
[12] S. Katti, S. Shintre, S. Jaggi, and D. K. M. Médard, "Real network codes," in *Proc. of Forty-Fifth Annual Allerton Conference on Communication, Control, and Computing*, UIUC, IL, USA, Sep. 2007.
[13] H. Park, N. Thomos, and P. Frossard, "Approximate decoding approaches for network coded correlated data," *Signal Processing (Elsevier)*, vol. 93, no. 1, pp. 109–213, Jan. 2013. Article (CrossRef Link)
[14] M. Kwon and H. Park, "An improved approximate decoding with correlated sources," *SPIE Optical Engineering + Applications. International Society for Optics and Photonics, San Diego*, CA, USA, Aug. 2011. Article (CrossRef Link)
[15] M. Kwon, H. Park, and P. Frossard, "Improved approximate decoding based on position information matrix," in *Proc. of IEEE Symposium on Computers and Communications*, Cappadocia, Turkey, Jul. 2012. Article (CrossRef Link)
[16] M. Kwon and H. Park, "Approximate recovery of network coded real-time information," in *Proc. of International Conference on Information Networking (ICOIN)*, Phuket, Thailand, Feb. pp. 545–549, 2014. Article (CrossRef Link)
[17] A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, and P. Gauthier, "Cluster-based scalable network services," *ACM SIGOPS Operating Systems Review*, vol. 31, no. 5, pp. 78–91, Oct. 1997. Article (CrossRef Link)
[18] J. Kim and J. Lee, "Cluster-based mobility supporting wmn for iot networks," in *Proc. of Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, Nov., pp. 700–703, 2012. Article (CrossRef Link)
[19] A. Mehmood, S. Khan, D. Zhang, J. Lloret, and S. H. Ahmed, "Iotec: Iot based efficient clustering protocol for wireless sensor network," in *Proc. of International conference on Industrial Information Systems*, Dec. 2014.
[20] H. Kim, J.-M. Chung, and C. H. Kim, "Secured communication protocol for internetworking zigbee cluster networks," *Computer Communications*, vol. 32, no. 13, pp. 1531–1540, 2009. Article (CrossRef Link)
[21] J.-M. Chung, S.-C. Kim, W.-C. Jeong, and S.-S. Joo, "Minimised power consuming adaptive scheduling mechanism for cluster-based mobile wireless networks," *Electronics letters*, vol. 45, no. 19, pp. 985–987, 2009. Article (CrossRef Link)
[22] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of IEEE 33rd Annual Hawaii International Conference on System Sciences*, 2000. Article (CrossRef Link)
[23] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002. Article (CrossRef Link)
[24] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004. Article (CrossRef Link)
[25] Feizi, Soheil, Muriel Médard, and Michelle Effros, "Compressive sensing over networks," in *Proc. of 48th Annual Allerton Conference on Communication, Control, and Computing*, 2010. Article (CrossRef Link)

[26] Rajawat, Ketan, Alfonso Cano, and Georgios B. Giannakis, "Network-compressive coding for wireless sensors with correlated data," *IEEE Transactions on Wireless Communications*, vol.11, no.12, pp. 4264-4274, 2012. Article (CrossRef Link)
[27] Maierbacher, Gerhard, Joao Barros, and Muriel Médard. "Practical source-network decoding," in *Proc. of IEEE 6th International Symposium on Wireless Communication Systems (ISWCS 2009)*, 2009. Article (CrossRef Link)
[28] Minhae Kwon, Hyunggon Park and Pascal Frossard, "Compressed Network coding: Overcome All-Or-Nothing Problem in Finite Field," in *Proc. of IEEE Wireless Communications and Networking Conference 2014 (WCNC 2014)*, Apr. 2014. Article (CrossRef Link)
[29] J.A Nelder, R.W.M Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society. Series A (General)*, vol. 135, no. 3, pp. 370-384, 1972. Article (CrossRef Link)
[30] Guisan, Antoine, Thomas C. Edwards, and Trevor Hastie, "Generalized linear and generalized additive models in studies of species distributions: setting the scene," *Ecological modelling*, vol. 157, no. 2, pp. 89-100, 2002. Article (CrossRef Link)
[31] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proc. of Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Oct. 2003.

Minhae Kwon received the BS and MS degrees in 2011 and 2013, respectively, and currently working toward Ph.D. degree in Department of Electronics Engineering, Ewha Womans University, Seoul, Korea. Her research interest includes robust strategies for delay-sensitive data transmission using network coding.

Hyunggon Park received the BS degree in electronics and electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, Korea, and the MS and Ph.D. degrees in electrical engineering from the University of California, Los Angeles (UCLA), in 2004, 2006 and 2008, respectively. Currently, he is an associate professor at the Department of Electronics Engineering, Ewha Womans University, Seoul, Korea. In 2009-2010, he was a senior researcher at the Signal Processing Laboratory (LTS4), Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. His research interests include game theoretic approaches for distributed resource management (resource reciprocation and resource allocation) strategies for multi-user systems and multi-user transmission over wireless/wired/peer-to-peer (P2P) networks, efficient and robust multimedia streaming strategies using network coding, fairness paradigms for resource management using game theory, big data processing for Internet of Things (IoT) and data stream mining based on machine learning, and cooperative resource management for 5G wireless networks. He was a recipient of the Graduate Study Abroad Scholarship from the Korea Science and Engineering Foundation during 2004-2006 and a recipient of the Electrical Engineering Department Fellowship at UCLA in 2008. He is a senior member of the IEEE.