

# 소규모 무선 센서 네트워크에서 분산 저장을 위한 LT 부호의 OBP 복호

정호영\*

## On-the-Fly Belief Propagation Decoding of Decentralized Luby Transform Codes for Distributed Storage in a Small-Scale Wireless Sensor Network

Ho-Young Cheong\*

**요약** 본 논문에서는 소규모 무선 센서 네트워크에서 데이터 저장을 위한 분산 LT 부호의 2 가지 복호 알고리즘의 성능을 시뮬레이션을 통해 분석하였다. 시뮬레이션 결과 복호율이 2.0 이상일 때 OBP 복호 알고리즘의 복호 성공 확률은 약 99%에 이르지만, BP 복호 방식의 경우  $n=100$ 일 때 약 50%를 하회하며  $n=200$ 일 때 약 70%에 이르는 것을 볼 수 있다. 따라서 소규모 무선 센서 네트워크에서 분산 저장을 위한 LT 부호의 복호 방식은 OBP 복호 방식이 크게 유리함을 알 수 있다.

**Abstract** In this paper I analyzed two decoding algorithms of decentralized LT codes for distributed storage by simulations in small-scale wireless sensor network. From the simulation results we can see that when the decoding ration is above 2.0, the successful decoding probability of OBP decoding is about 99%, while that of BP decoding is below 50% with  $n=100$ , and about 70% with  $n=200$ . We showed that OBP decoding algorithm is an efficient decoding scheme for decentralized LT codes for distributed storage in small-scale wireless sensor network.

**Key Words** : Belief Propagation, On-the-Fly Belief Propagation, Triangularization, Query Ratio, Decentralized Luby Transform Code.

### 1. 서론

무선 센서 네트워크는 한정된 자원을 가진 많은 센서 노드들로 구성된다. 센서 노드들이 위치한 곳은 대개 인간의 손길이 닿지 않는 척박한 환경에 있는 경우가 많아 센서 노드들은 한정된 에너지를 가지고 데이터를 간단히 처리하거나 통신 장치를 통해 안전한 이웃 노드에 데이터를 보내야 한다 [1]. 따라서 센서 노드들에 의해 발생하는 데이터

들은 대부분 수명이 짧으며 데이터를 더 복잡한 형태로 처리하거나 외부로 안전하게 전송하기 위해서는 강력한 데이터 처리 능력과 안정된 전원 및 통신 능력을 가진 싱크 노드 혹은 데이터 수집 장치 등이 있어야 한다[2]. 이러한 경우 획득한 정보 데이터들을 네트워크 내에 분산 저장하여 노드들이 유실되는 경우에도 데이터 수집 장치가 다른 노드들로부터 유실된 노드 데이터를 복구할 수 있도록 하는 것은 매우 중요하다[1][2].

Funding for this paper was provided by 2015 Internal Research Project of Namseoul University

\*Corresponding Author : Department of Information and Communication Engineering, Namseoul University (hycheong@nsu.ac.kr)

Received October 09, 2016

Revised October 25, 2016

Accepted October 25, 2016

$n$  개의 노드들로 구성된 무선 센서 네트워크에서  $k$  개의 센서 노드들이 각각 독립적인 정보를 발생시킨다고 가정할 때 이들 데이터를 네트워크 내에 분산시켜  $n$  개의 센서 노드들이 부호화된 패킷 하나씩을 분산 저장하는 것이 바람직하다. 이를 위해 임의의  $k(1+\epsilon)$  개의 노드들로부터 원래의  $k$  개 정보 패킷들을 복원할 수 있는 분산 저장 부호에 대한 수많은 연구가 진행 되어 왔다. Dimakis 등은 무선 센서 네트워크에서 데이터 분산 저장을 위해 분산 부호화(decentralized coding) 형태의 파운틴 부호 알고리즘을 처음 제안하였으며[3], 최근에 Z. Kong 등은 LT 부호(Luby transform code)에 기반한 분산 저장 부호를 제안하였다. 이들이 제안한 알고리즘은 무선 센서 네트워크의 노드들이 네트워크 전역 정보 변수인  $n$ ,  $k$  값은 물론 라우팅 테이블 정보도 전혀 모른다는 상황 하에서 부호화가 이루어지며 기존의 LT 부호에 근접한 성능을 보여 주고 있다[4]. D. Vukobratović 등은 네트워크 내의 분산된 노드에서 부호화하는 것이 아니라 무부호율 패킷(rateless packet)이라고 부르는 원시 부호 패킷이 네트워크 내의 분산된 센서 노드들을 랜덤하게 방문하여 이들이 가지고 있는 정보데이터들을 부호화하는 패킷 중심(packet-centric) 분산 저장 부호 알고리즘을 제안한 바 있다[1]. 한편 Kamra 등은 무선 센서 네트워크상의 데이터 생존기간(persistence)을 늘리기 위해 성장 부호화(growth coding)라고 부르는 새로운 기법을 제안하였다[5].

위에서 말한 모든 기법들은 기본적으로 대규모 무선 센서 네트워크를 가정하고 있으나 최근 센서 노드들의 수가 수십 개에서 수백 개 이내로 제한된 ad-hoc 형태의 소규모 무선 센서 네트워크들이 자주 발생하고 있다[6]. 본 논문에서는 소규모 무선 센서 네트워크에서 데이터 분산 저장을 위한 분산 LT 부호의 복호 성능을 분석하고 시뮬레이션을 통해 효율적인 복호 알고리즘을 제시한다. 특히 Z. Kong 등에 의해 제시된 LTCDS(LT codes based distributed storage) 부호의 BP(belief propagation) 복호 기법과 본 논문에서 제시하는

OBP(on-the-fly belief propagation) 복호 기법의 성능을 소규모 무선 센서 네트워크의 상황 하에서 시뮬레이션을 통해 비교·고찰한다. 본 논문의 구성은 다음과 같다. 제 2 장에서는 무선 센서 네트워크를 랜덤 그래프로 모델링하였으며, 제 3 장에서는 무선 센서 네트워크상에서 데이터 분산 저장을 위한 LT 부호의 부/복호 알고리즘을 기술한다. 제 4 장에서 이들에 대해 시뮬레이션을 통해 부호 패킷 복호율 및 네트워크 파라미터에 따른 복호 성공 확률을 복호 알고리즘 별로 시뮬레이션을 통해 실험한 결과를 비교·분석하고 제 5 장에서 결론을 맺는다.

## 2. 랜덤 그래프를 이용한 무선 센서 네트워크 모델링

무선 센서 네트워크는  $n$  개의 노드들을 갖는 랜덤 그래프로 모델링할 수 있으며 노드들은 평면 상에서 균일하고도 랜덤하게 분산되어 있다고 가정한다. 모든 노드들은  $r$ 의 통신 반경을 가지며 반경  $r$  이내의 거리에 위치해 있는 노드들끼리는 서로 안전하게 통신이 가능하며 이러한 두 노드들을 이웃 노드(neighbor)라고 한다[4].  $n$  개의 노드들 중에서 센싱된 데이터를 발생시키는  $k$  개의 소스 노드들이 존재하며 이들로부터 독립적으로 발생된 정보는 저장을 위해 네트워크를 통해 분산 저장된다. 본 논문에서는 소규모 무선 센서 네트워크를 대상으로 하고 있으므로 모든 노드들은 네트워크의 구성 정보인  $n$ ,  $k$  값을 미리 알고 있다고 가정한다. 본 논문에서 가정하는 무선 센서 네트워크는 전체 노드 수가  $n \leq 500$  이하이고 소스 노드의 수는  $k = 0.1n \sim 0.2n$ 의 값을 갖는 소규모 무선 센서 네트워크이다. 앞으로 본 논문의 표기법은 [4]의 표기법을 준용하기로 한다.

$k$  개의 소스 노드로부터 발생된  $k$  개의 정보 데이터를 분산 저장하기 위해 데이터 패킷은 랜덤 워크(random walk) 형태로 분산 전송된다. 각 노드에서 랜덤 워크 패킷을 수신하면 부호화 과정을 수행하여 부호화된 부호 패킷을 저장하고 랜덤 위

크 패킷은 다시 이웃 노드 중 하나로 전송된다.  $k$  개의 랜덤 워크 패킷들이 랜덤 그래프 내의 노드들을 최소 1회 이상 방문할 수 있도록 하기 위해 랜덤 워크 길이는 일정 시간(네트워크 커버 시간) 이상 지속해야 한다. 랜덤 그래프  $G$ 의 네트워크 커버 시간  $T_{cover}(G)$ 는 식 (1)과 같이 정의된다.

$$T_{cover}(G) = \text{Max}_{u \in G} T_{cover}(u) \quad (1)$$

여기에서  $T_{cover}(u)$ 는 노드  $u$ 를 출발하여 그래프  $G$  안에 있는 모든 노드들을 최소 한 번 이상 방문하는 랜덤 워크의 평균 길이이다. 또한 Avin과 Ercal[7]의 결과로부터  $T_{cover}(G)$ 는 식 (2)와 같이 놓을 수 있다.

$$T_{cover}(G) = \Theta(n \log n) \quad (2)$$

따라서 임의의 랜덤 워크 길이가  $C_1 n \log n$  ( $C_1 > 0$ )이면 랜덤 그래프의 각 노드들을 1회 이상 방문할 수 있다고 확신할 수 있다.  $k$  개의 랜덤 워크들이 위의 조건을 충족하도록 각 소스 패킷에 카운터를 장착하여 매 전송 때마다 카운터 값을 증가시킨다. 카운터 값이  $C_1 n \log n$  보다 작은 패킷을 수신할 때 마다 노드는 부호화 과정을 수행한 후 다음 노드를 랜덤하게 선택하여 패킷을 전송하고, 수신 패킷의 카운터 값이 ( $C_1 n \log n$ )보다 큰 패킷은 버린다.  $k$  개의 소스 패킷들이 모두  $C_1 n \log n$  값 보다 큰 값을 가져 수명을 다하면 분산 전송(dissemination) 과정은 종료된다.

### 3. 분산 저장을 위한 LT 부호의 부/복호 알고리즘

$k$  개의 소스 패킷들을

$\mathbf{m} = [m_0, m_1, \dots, m_{k-1}]$ 으로 표시하면  $k$  개의 소스 패킷들을 한 곳에 모아 부호화기를 통해

부호 패킷  $\mathbf{y} = (y_1, y_2, \dots)$ 를 만든 후 전송되는 것이 일반적인 LT(Luby transform) 부호이다. 부호 패킷  $y_i$ 는  $k$  개의 소스 패킷들로부터 식 (3)과 같이 생성된다[8][9].

$$y_i = m_{r_0} \oplus m_{r_1} \oplus \dots \oplus m_{r_{d-1}} \quad (1)$$

부호화기는  $[0, k]$  범위에 있는 정수  $d$ 를 RSD(robust Soliton distribution) 확률 분포  $\Omega$ 를 이용하여 생성한 후  $d$  개의 소스 패킷들을 랜덤하게 선택하여 식 (3)과 같이 XOR 연산을 적용하여 부호 패킷을 만든다. 모든 부호 패킷은 부호 패킷을 생성하기 위해 사용한  $d$  개의 소스 패킷들이 XOR 연산에 사용되었는지의 여부를 나타내는 정보 벡터  $\mathbf{b}_i = [b_{i,0}, b_{i,1}, \dots, b_{i,k-1}]$ 와 함께 전송된다. 이때 소스 패킷  $m_j$ 가 부호 패킷  $y_i$ 를 생성하는데 사용되었으면  $b_{i,j} = 1$  값을 갖고 사용되지 않았으면  $b_{i,j} = 0$ 의 값을 가진다. 이때  $m_{r_j}$ 는

XOR 연산에 사용된  $r_j$  번째 메시지 소스 패킷을 의미한다.

#### 3.1 분산 저장을 위한 LTCDS 부호화 알고리즘

기존의 LT 부호의 부호화 과정은  $k$  개의 소스 데이터를 한 곳에서 모두 모아 놓은 후 이들로 부터 확률 분포  $\Omega$ (robust Soliton 혹은 ideal Soliton 분포)를 이용하여 랜덤하게 선정한  $d(\leq k)$  개의 소스 데이터들에 대해 XOR 연산을 수행하여 LT 부호 패킷을 생성한다. 그러나 본 논문에서 수행하고자하는 분산 저장 LT 부호의 부호화는 무선 센서 네트워크상에  $k$  개의 소스 노드가 랜덤하게 분산되어 있으므로 데이터 패킷을 한 곳에 모아 부호화를 할 수 없다. 따라서 흩어져있는  $n$  개의 노드들이 각자  $k$  개의 소스 데이터에 대한 LT 부호 패킷을 생성하여 분산된 형태로 저장하는 것이다. 이를 위해 네트워크 내의 각 노드들은 확률 분포  $\Omega(d)$ 를 이용해  $d$ 를 발생시킨 후 랜덤 워크 형태로 방문하는 소스 데이터 패킷 중에서  $d$  개의 소

스 패킷을 이용하여 부호화한 후 부호 패킷을 각 자 저장해야 한다. Zhenning Kong 등에 의해 제안된 LTCDS(LT codes based distributed storage) 부호화 알고리즘을 위의 분산 부호화 과정을 충실히 구현하고 있으므로 [4]의 부호화 과정을 이용하여 복호 성능을 시뮬레이션을 통해 분석한다. 분산 형태의 LT 부호화에 사용되는 확률 분포  $\Omega$ 는 RSD를 사용된다고 가정하면 다음과 같이 요약할 수 있다.

먼저 초기화 단계로 각 소스 노드  $s(s=1, \dots, k)$ 에 대해 소스 데이터  $x_s$ 에  $x_s$ 의 ID와 수명 카운터 등을 포함하고 있는 헤더를 장착한다. 각 소스 노드는 랜덤하게 선정된 이웃 노드에 소스 패킷을 보내고, 저장 노드  $u$ 는 저장 부호를  $y_u=0$ 으로 초기화한다.

둘째로, 부호화와 저장 단계로 임의의 노

드  $u$ 에서 확률 분포  $\Omega$ 에 따라 값  $d$ 를 생성한다. 랜덤 워크로 도착한 패킷에 대해 카운터 값이  $C_1 n \log n$  보다 작으면 노드  $u$ 는 노드  $u$ 는  $(d/k)$ 의 확률로 도착한 소스 패킷을 부호화 대상 패킷으로 선택하여 부호화한다. 수신 패킷의 카운터 값을 증가시키고 부호화 과정이 끝나면 다음 이웃 노드로 전송할 수 있도록 대기시킨다. 만일 카운터 값이  $C_1 n \log n$  이상이면 랜덤 그래프 내의 모든 노드들을 1회 이상 방문한 것으로 보아 제거한다. 각 노드가  $k$  개의 소스 패킷을 수신했을 경우 부호화 과정을 종료한다.

### 3.2 데이터 분산 저장을 위한 복호 알고리즘

네트워크상에 랜덤하게 분산되어 있는  $n$  개의 저장 노드들에 대해 데이터 수집 장치는 임의의  $k(1+\epsilon)$  개의 노드들에게 각 노드가 저장하고 있는 부호 패킷을 요청하여 원래의  $k$  개 정보 패킷들을 복호한다. 이때  $k$  개의 소스 패킷을 복호하기 위해 부호 패킷을 요청한 노드들의 수  $h=k(1+\epsilon)$ 와  $k$ 의 비율을 복호율(decoding ratio)  $\eta(\eta=h/k)$ 이라고 한다. 데이터 수집 장치에 전송되는 부호 패킷에는 부호화에 사용된 소스 패킷

정보가 담긴 정보 벡터  $\mathbf{B} = [b_1, b_2, \dots, b_n]^T$ 가 같이 수신되며 이를 이용하면 선형 방정식  $\mathbf{B}\mathbf{m} = \mathbf{y}$ 을 구성할 수 있다. 여기에서 시스템  $\mathbf{B}\mathbf{m} = \mathbf{y}$ 로부터 독립이 아닌 등식들을 제거하면  $\mathbf{B}$ 와  $\mathbf{y}$ 로부터 각각  $(k \times k)$  행렬  $\mathbf{G}$ 와  $(k \times 1)$  벡터  $\mathbf{Y}$ 를 얻을 수 있고  $\mathbf{G}\mathbf{m} = \mathbf{Y}$ 를 풀어  $\mathbf{m}$ 을 복호할 수 있다.

#### 3.2.1 BP(belief propagation) 복호 알고리즘

BP 복호 알고리즘에서는 차수가 1인 행  $\mathbf{G}[i]$ 를 탐색하여 행  $\mathbf{G}[i]$ 에서 '1'의 열 위치가  $j$ 라고 하면  $m_j = y_i$ 로 복호한다. 이때  $\mathbf{G}[i]$ 는  $\mathbf{G}$  행렬의  $i$  번째 행을 의미한다. 동시에  $\mathbf{G}[i]$ 의  $j$  번째 열에 있는 모든 '1'들을 '0'으로 바꿈과 동시에 '0'으로 바뀌는 행에 해당하는 부호 패킷  $y_l$ 을  $y_i$ 와 XOR 연산을 하여 선형 방정식을 갱신한다. BP 복호 알고리즘에서는  $\mathbf{G}$ 의 행 벡터  $\mathbf{G}[i]$ 의 차수를 탐색하여 차수가 1인 행벡터를 저장해 놓았다가 한꺼번에 복호하는데 이때의 저장 버퍼를 리플(ripple)이라고 한다. 리플에 저장된 차수 1의 행벡터를 이용해 복호하고 아울러 행렬  $\mathbf{G}$ 를 갱신하는 것을 리플을 푼다고 하며, 갱신된 행렬  $\mathbf{G}$ 로부터 다시 차수가 1인 행을 찾아 리플 원소를 채운 후 위와 같은 복호 과정을  $\mathbf{G}$  행렬의 원소가 모두 0이 될 때까지 반복한다.  $\mathbf{G}$  행렬의 원소가 모두 0이 되면 복호가 성공한 것이고 모두 0이 되지 않았는데도 불구하고 차수가 1인 행이 없으면 복호 과정이 중단되어 실패하게 된다.  $\mathbf{G}$  행렬의 원소가 모두 0이 되지 않았음에도 불구하고 리플이 비게 되면 주위의 노드들에게 부호 패킷을 요청하게 되고 차수가 1인 부호 벡터를 기다려 복호를 재개하게 된다. 이러한 이유로 BP 복호 알고리즘은  $k$  값이 작을 때 리플이 자주 비게 되고 차수 1의 부호 패킷이 도착할 때 까지 다른 패킷을 사용하지 못하므로 오버헤드가 증가한다.

3.2.2 OBP(on-the-fly belief propagation) 복호 알고리즘

OBP 복호 알고리즘에서는  $k(1+\epsilon)$  개의 노드들에게 요청하여 수신한 부호 패킷 중에서  $k$ 개의 부호 패킷이 도착하면 BP 복호 알고리즘과 동일하게 복호한다. 리플이 비게 되면 BP 알고리즘과 달리 도착하는 부호 패킷의  $b_i$  행렬 차수가 1이 아니어도 기존에 있는 행렬  $G$ 의 삼각화(triangularization) 과정을 통하여 새로운 차수 1의 행렬을 만든다. 이 과정에서 XORing 연산과 swapping 과정이 사용된다. 이들 두 과정은 행렬  $G$ 를 구성하는  $G[i]$ 의 차수를 감소시켜 요청해 받은 행 벡터는 아니지만 자체적으로 차수 1의 행 벡터를 얻게 되는 중요한 역할을 하며, 복호 과정이 끝날 때까지 리플에 차수-1의 원소들을 충분히 공급해 주는 중요한 기능을 가지고 있다. 특히 위의 과정에서 삼각화가 완성되면 후향 대입법을 통해 나머지 모든 해를 구할 수도 있다.

본 논문에서는 삼각화와 BP 복호 알고리즘을 동시에 진행하는 OBP 복호 알고리즘을 무선 센서 네트워크의 분산 저장 부호에 대한 복호 방식으로 제안한다. 기존의 BP 복호 방식은 복호 도중 리플이 비어 복호과정이 중단될 경우 새로운 차수-1의 패킷이 공급되지 않는 한 실패하게 된다. 특히  $k$  값이 작을 경우 많은 오버헤드가 발생하여 복호 효율이 급격히 낮아진다. 그러나 OBP 복호 알고리즘에서는 차수-1의 패킷이 없어도 차수-2 이상의 패킷들에 대해 XORing 및 swapping 연산을 하여 차수-1을 갖는 패킷을 생성할 수 있어 복호 과정이 중단되지 않는 것은 물론 오버헤드를 줄일 수 있다. 그러나 이는  $k$  값이 작을 때 적용할 수 있으며  $k$  값이 커지면 복호 연산량은 기하급수적으로 증가하여 실현성이 떨어지게 된다. 삼각화 과정의 swapping은 단순히 비트열  $b_i$ 와  $b_j$ 를 교환하는 것이므로 연산 복잡도는 증가하지 않으며 XORing 연산에 의해서만 복잡도가 증가한다. 본 논문에서 제안한 OBP 복호 알고리즘을 요약하면 다음과 같다[10].

- (1)  $k$  개의 패킷을 수신한 후 기존의 BP 복호 알고리즘을 적용하여 복호한다.
- (2) 리플이 비어 복호 과정이 중단되면 새로운 패킷 정보  $b_j$ 를 수신하여 행렬  $G$ 에 삽입하고 이미 복호된 비트 위치를 0으로 바꾼다.
- (3) 행렬  $G$ 의 행 벡터  $b_i$ 와 수신 벡터  $b_j$ 의 열이 좌측으로부터 동일한 위치에 1의 값을 가지면  $b_k = b_i \oplus b_j$  XORing 연산을 적용한다. XORing 연산 시  $W(b_j) \leq W(b_i)$  이면 swapping을 수행한다. 단,  $W(b_i)$ 은  $b_i$  벡터에 존재하는 1의 개수이다.
- (4) XORing 연산과 swapping 과정이 종료된 후  $b_k$ 가 전영 벡터가 아니면  $B$  행렬에 저장한다.
- (5) 차수-1의 패킷을 탐색하여 BP 복호를 재개한다. 복호가 종료되지 않았는데도 차수-1의 패킷이 없으면 (2)~(5)의 과정을 반복한다.

4. 시뮬레이션 및 결과 분석

본 논문에서는  $n \leq 500$ 인 소규모 무선 센서 네트워크에서 데이터를 분산 저장하기 위한 LT 부호의 복호 성능을 살펴보기 위해 다음과 같이 시뮬레이션을 수행하였다.

무선 무선 센서 네트워크는 랜덤 워크를 통해 데이터가 전송되는 랜덤 그래프로 모델링하였다. 노드 연결성을 확보하기 위해 확률  $p=0.03$ 의 값을 갖는 Erdos-Renyi 알고리즘을 이용하여  $n$  개의 노드로 구성되는 랜덤 그래프를 생성하였다. 생성된 랜덤 그래프 상에서  $n$  개의 노드들 중에서 약 10~20%에 해당하는  $k$  개의 소스 노드를 균일하고도 랜덤한 방식으로 선정하여 소스 노드로 설정하였다. LT 부호화에 사용되는 확률 분포는 RSD(robust Soliton distribution) 확률 분포  $\Omega$ 를 사용하여 부호화 패킷 수  $d$ 를 생성하였으며 이 때  $\Omega$ 를 구성하는 파라미터는  $c=0.1$ 과  $\delta=0.01$ 로 하였다.

데이터 수집 장치가 복호를 위해 부호 패킷을 요청한 주변의 네트워크 노드들이  $h$ 개라고 하면 복호율  $\eta$ 는  $\eta = h/k$ 로 계산할 수 있다.  $n$  개의 노

드들 중에서  $h$  개의 노드들을 랜덤하게 선택하여 데이터 요청 노드 집합을 구성한 후 BP 복호와 OBP 복호 알고리즘을 적용하여 시뮬레이션을 수행하였다.

그림 1은 네트워크의 총 노드 수가  $n=100$ 일 때 복호율  $\eta$ 에 따른 복호 성공률  $P_s$ 를 나타낸 것이다. 그림 1에서 OBP 복호 방식의 경우 복호율이  $\eta \geq 2.0$ 인 곳에서 복호 성공 확률이 거의 99%에 근접하나 BP 복호 방식의 경우 약 50%에 머물고 있음을 알 수 있다. 또한 BP 복호 방식과 OBP 복호 방식 모두  $\eta \leq 1.3$ 일 때 총 노드 수  $n$ 에 대한 소스 노드 수  $k$ 가 클수록 복호 성공 확률은 떨어짐을 알 수 있다.

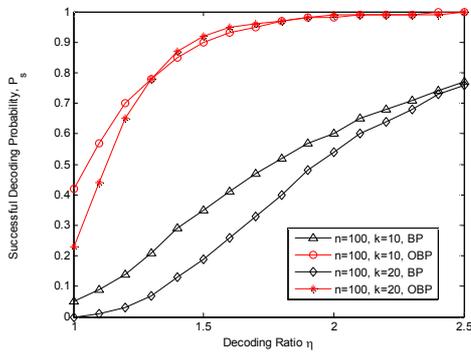


그림 1. 복호율에 따른 BP 복호 방식과 OBP 복호 방식의 복호 성공 확률( $n=100, C_1=5$ )

Fig. 1. Successful Decoding Probability of BP and OBP decoding schemes vs  $\eta$  ( $n=100, C_1=5$ )

그림 2는 네트워크의 총 노드 수가  $n=200$ 일 때 복호율  $\eta$ 에 따른 BP 복호 방식과 OBP 복호 방식의 복호 성공률  $P_s$ 를 나타낸 것이다. 그림 1과 비교해 볼 때  $k=0.1n$ 인 BP 복호 방식은  $n=100$ 일 때에 비해 복호 성공 확률이 크게 상승하여  $\eta=2.0$ 일 때 약 70%에 육박하는 값을 보여 준다. 그러나  $k=0.2n$ 인 경우에는 여전히 50% 미만의 성공 확률을 보여 준다. 이는 리플이 빈 경우

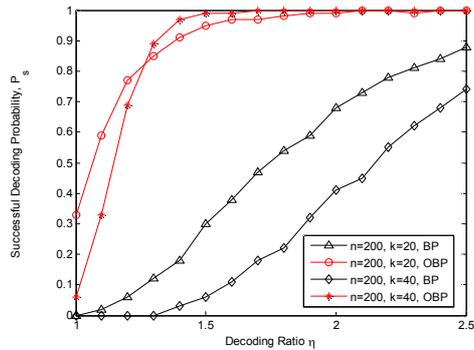


그림 2. 복호율에 따른 BP 복호 방식과 OBP 복호 방식의 복호 성공 확률( $n=200, C_1=5$ )

Fig. 2. Successful Decoding Probability of BP and OBP decoding schemes vs  $\eta$  ( $n=200, C_1=5$ )

차수 1의 부호 패킷을 저장하고 있는 노드 수가 상대적으로 빈약하기 때문인 것으로 분석된다. OBP 복호 방식의 복호 성공 확률은 더욱 양호한 값을 보여주며 복호율이  $\eta \geq 2.0$ 인 곳에서는 복호 성공 확률이 거의 100%에 근접함을 알 수 있다.

특히  $k=0.2n$ 인 OBP 복호 방식의 경우  $\eta \geq 1.5$ 인 곳에서 급격히 100%에 근접하는데 이는  $k$  값이 클수록 리플이 비었을 때 차수 1의 부호 패킷을 자체적으로 발생시킬 수 있는 확률이 커지기 때문이다.

### 5. 결론

본 논문에서는 소규모 무선 센서 네트워크에서 데이터 분산 저장을 위한 분산 LT 부호의 복호 성능을 분석하고 시뮬레이션을 통해 효율적인 복호 알고리즘을 제시하였다.

확률  $p=0.03$ 의 값을 갖는 Erdos-Renyi 알고리즘을 이용하여  $n$  개의 노드로 구성되는 랜덤 그래프를 생성하였고, 복호율에 따른 BP 복호 방식과 OBP 복호 방식의 복호 성공 확률을 측정하여 분석하였다.

시뮬레이션 결과  $n=100$ 일 때 OBP 복호 방식의 경우 복호율이  $\eta \geq 2.0$ 인 곳에서 복호 성공 확

률이 거의 99%에 근접하나 BP 복호 방식의 경우 약 50%에 머물고 있음을 알 수 있었다.  $n=200$ 일 때  $k=0.1n$ 인 BP 복호 방식은  $n=100$ 일 때에 비해 복호 성공 확률이 크게 상승하여  $\eta=2.0$ 일 때 약 70%에 육박하는 값을 보여준다. 그러나  $k=0.2n$ 인 경우에는 여전히 50% 미만의 성공 확률을 보여 준다. 이는 리플이 빈 경우 차수 1의 부호 패킷을 저장하고 있는 노드 수가 상대적으로 빈약하기 때문인 것으로 분석된다.

$k$  값이 크지 않은 소규모 무선 센서 네트워크에서 분산 LT 부호의 복호 방식은 OBP 복호 방식이 BP 복호 방식에 비해 크게 유리함을 알 수 있었다. 그러나 OBP 복호 방식은  $k$  값이 커지면 복호 연산 량이 크게 증가하는 특성을 가지고 있으므로, 향후  $k$  값에 따른

복잡도(연산량)와 복호 성공 확률의 관계를 분석하는 것이 필요하다.

### 감사의 글

이 논문은 2015년도 남서울대학교 학술연구비 지원에 의해 연구되었음

### REFERENCES

- [1] Dejan Vukobratovič, Čedomir Stefanovič, Vladimir Cmojevič, Francesco Romano Fantacci, "Rateless Packet Approach for Data Gathering in Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications* Volume: 28, Issue: 7, September 2010
- [2] C. Raghavendra, K. Sivalingam, and T. Znati, *Wireless Sensor Networks*. New York: Springer, 2004.
- [3] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Distributed fountain codes for networked storage," in *Proc. of 31st IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP' 06)*, Toulouse, France, May, 2006.
- [4] Zhenning Kong, Salah A. Aly, Emina Soljanin, "Decentralized Coding Algorithms for Distributed Storage in Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, Volume: 28, Issue: 2, February 2010.
- [5] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes: Maximizing sensor network data persistence," in *Proc. of ACM Sigcom' 06, Pisa, Italy*, September, 2006.
- [6] Samira Chouikhi, Inès El Korbja, Yacine Ghamri-Doudanec, Leila Azouz Saidane, "A Survey on Fault Tolerance in Small and Large Scale Wireless Sensor Networks," *Computer Communications*, Volume 69, 15 September 2015, pp. 22 - 37.
- [7] C. Avin and G. Ercal, "On the cover time of random geometric graphs," in *Proc. 32nd International Colloquium of Automata, Languages and Programming (ICALP' 05)*, Lisboa, Portugal, pp. 677 - 689, July, 2005.
- [8] M. Luby, "LT Codes," *Proc. of the 43rd Annual IEEE Symp. on Foundations of Comp. Sc.*, pp 271-280, Vancouver, Canada, November 2002
- [9] Valerio Bioglio, Marco Grangetto, Rossano Gaeta, Matteo Sereno, "On the fly Gaussian Elimination for LT Codes," *IEEE Communications Letters*, Volume: 13, Issue: 12, December 2009, pp 935-955.
- [10] Hoyoung Cheong, "An Improved Belief Propagation Decoding for LT Codes," *Journal of KIIECT*, Vol.7 No. 4, Dec. 2014, pp. 223~228.

---

저자약력

---

정 호 영(Ho-Young Cheong)

[중신회원]



- 1987년 8월 : 연세대학교 대학원 전자공학과 (공학석사)
- 1995년 2월 : 연세대학교 대학원 전자공학과 (공학박사)
- 1995년 4월 ~ 현재 : 남서울대학교 정보통신공학과 교수

<관심분야>

채널부호, 분산 저장 부호, 무선 센서 네트워크