LT 부호를 위한 개선된 BP 복호

정호영*

An Improved Belief Propagation Decoding for LT Codes

Ho-Young Cheong*

요 약 LT 부호에 대해 BP 복호 알고리즘은 가장 빠른 복호 방법 중 하나로 알려져 있다. 그러나 BP 알고리즘은 대부분의 LT 부호를 복호하는데 있어서 많은 오버헤드를 요구하며 특히 짧은 길이의 LT 부호에 대해서는 과다한 오버헤드가 소요된다. 본 논문에서는 오버헤드를 줄이기 위해 1-차수의 패킷을 탐색할 수 있는 방법을 제시하고 이를 이용한 개선된 BP 복호 알고리즘을 제안하였다. 제안된 복호 알고리즘은 기존의 BP 알고리즘에 비해 같은 복호 복잡도를 유지하면서도 더 적은 오버헤드를 가짐을 알 수 있었다.

Abstract It is known that a belief propagation algorithm is a fast decoding scheme for LT codes but it require a large overhead, especially for a short block length LT codes. In this paper an improved belief decoding algorithm using searching method for degree-1 packets is proposed for a small overhead. The proposed decoding scheme shows the desirable performance in terms of overhead while guaranteeing the same computational complexity with respect to the conventional BP decoding scheme.

Key Words : Belief Propagation, LT code, Overhead, On-the-Fly Gaussian Elimination, Xoring,

1. 서론

패킷 전송을 하는 인터넷 망은 BEC(binary erasure channel) 채널의 대표적인 형태로 꼽힌다[1]. 인터넷 망을 통해 패킷이 전송되는 경우 패킷 내에 오류가 발생하기 보다는 전송되는 과정에서 패킷이 유실되어(packet loss) 없어지는 경우가 대부분이므로 유선 인터넷 망은 BEC로 모델링하는 것이 적합하다. LT(Luby Transform) 부호는 BEC에서 채널 용량에 근접하는 성능을 보이는 부호로 부호 율(code rate)이 정해져 있지 않은 무부효율 부호로써 특히 멀티캐스팅 통신에 적합한 부호로 보고가 되고 있다[1].

LT 부호는 전송해야 할 k 개의 메시지 패킷들이 있을 경우 메시지 패킷들을 선형 조합하여 끊임없이 부호 패킷(code packet)들을 발생시켜 전송하게 되며 수신 단에서는 패킷의 전송 순서에 상관없이 원래의 메시지 패킷이 완전히 복원될 때까지 부호 패킷들을 랜덤하게 수신하여 복호한다. 특히 부호 패킷을 조합하는데 사용된 메시지 패킷들의 위치 정보를 담은 비트 열을 정보 벡터(information vector)라고 하는데, 송신 단에서는 부호 패킷을 정보 패킷과 함께 전송한다[2].

LT 부호를 위한 복호 기법에는 BP(belief propagation) 복호 기법이 대표적인데 다른 복호 방식에 비해 연산 복잡도가 가장 낮은 것으로 알려져 있다. 그러나 전송하고자 하는 부호 길이 k 가 작으면 작을수록 오버 헤드(overhead)가 지나치게 증가하는 단점을 가지고 있다. 또 다른 LT 부호의 복호 기법으로 GE(Gaussian elimination)

* Corresponding Author : Department of Information and Communication Engineering Professor of Namseoul University (hycheong@nsu.ac.kr)
Received : November 02, 2014 Revised : November 19. 2014 Accepted : December 2. 2014

복호 알고리즘은 복호 속도가 BP 알고리즘에 비해 느리고 연산 량이 많기는 하지만 오버헤드가 거의 없다는 장점을 가지고 있다. BP 복호 기법과 GE 복호 기법은 LT 부호의 기본적인 복호 기법으로 오버헤드와 연산 복잡도 면에서 서로 상충관계(trade-off)를 가지고 있다.

한편 OFG(On-the-Fly Gaussian Elimination) 복호 알고리즘은 GE 복호 알고리즘의 일종으로 일정 개수의 수신 패킷들이 도착할 때까지 기다리지 않고 처음부터 패킷들이 도착하자마자 복호 과정을 수행하는 특징을 가지고 있다. 패킷들이 도착하는 모든 시간대로 GE의 복호 연산과정을 분산시켜 복호를 수행함으로서 마지막 패킷이 도착한 후 짧은 시간 내에 복호 과정을 종료할 수 있기 때문에 전체적인 복호 시간을 보면 BP 복호 알고리즘이나 GE 복호 알고리즘 보다 더 짧다[3].

본 논문에서는 연산 복잡도를 증가시키지 않으면서도 BP 복호의 단점인 오버헤드를 줄이기 위해 OFG 복호 기법의 Xoring을 이용한 개선된 BP 복호 알고리즘을 제안한다. 본 논문의 구성은 다음과 같다. 제 2 장에서는 LT 부호의 부/복호화 과정을 살펴보고, 제 3 장에서는 본 논문에서 제안한 xoring 연산을 이용한 BP 복호 알고리즘을 자세히 기술한다. 제 4 장에서 이들에 대해 시뮬레이션을 통해 오버헤드 비중과 복호 연산양을 중심으로 한 실험 결과를 비교·분석하고 제 5 장에서 결론을 맺는다.

2. LT 부호의 부/복호화 과정

전송하고자 하는 임의의 k 개 메시지 소스 패킷을 $\boldsymbol{m} = (m_0, m_1, \ldots, m_{k-1})$ 로 표시하자. LT 부호는 $\boldsymbol{m}$의 원소 패킷 들을 선형 조합하여 부호 패킷 $\boldsymbol{y} = (y_1, y_2, \ldots)$ 을 끊임없이 생성하여 전송한다. k 개의 메시지 패킷들을 이용해 수행하는 LT 부호의 부호화 과정은 다음과 같다[4].

(1) d 개의 랜덤 정수 $r_0, r_1, \ldots, r_{d-1}$을 RSD(robust soliton distribution) 분포를 이용해 구간 $[0, k-1]$에서 발생시킨다.

(2) k 개의 메시지 패킷 중에서 임의로 d 개를 선택한 후 xor 연산을 수행하여 부호화된 패킷 y_i를 식 (1)과 같이 얻는다.

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

여기에서 m_{r_j}는 xor 연산에 사용된 r_j 번째 소스 패킷을 의미한다.

(3) y_i의 xor 연산에 사용된 소스 패킷 정보를 담고 있는 정보 벡터 $b_i = [b_{i1}, b_{i2}, \ldots, b_{ik}]$ 와 함께 y_i를 전송하게 된다. 여기에서 m_j가 y_i를 생성하기 위해 xor 연산에 사용되었을 경우 $b_{ij} = 1$ 값을 갖고 사용되지 않았을 경우 $b_{ij} = 0$ 의 값을 갖게 된다.

수신단의 복호기가 n 개의 부호화된 패킷 $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^T$ 과 해당 위치 정보 벡터 $\boldsymbol{B} = [b_1, b_2, \ldots, b_n]^T$ 를 수신하게 되면 선형 방정식 $\boldsymbol{B}\boldsymbol{m} = \boldsymbol{y}$ 의 해를 구하여 소스 패킷 $\boldsymbol{m}$을 복호하게 된다. 여기에서 시스템 $\boldsymbol{B}\boldsymbol{m} = \boldsymbol{y}$ 로부터 독립이 아닌 등식들을 제거하면 $\boldsymbol{B}$와 $\boldsymbol{y}$로부터 각각 $(k \times k)$ 행렬 $\boldsymbol{G}$와 $(k \times 1)$ 벡터 $\boldsymbol{Y}$를 얻을 수 있고 $\boldsymbol{G}\boldsymbol{m} = \boldsymbol{Y}$를 풀어 m을 복호할 수 있다. 방정식 $\boldsymbol{G}\boldsymbol{m} = \boldsymbol{Y}$의 해는 GE(Gaussian elimination) 알고리즘을 이용해 구할 수 있다[5]. 그러나 GE를 이용할 경우 k 값에 따라 연산양이 급격히 증가하는 단점을 가진다. LT 부호는 벡터 b_i의 차수가 RSD 특성을 갖도록 적절히 조절하여 수신 단에서 GE 복호 알고리즘 대신 BP 복호 알고리즘이 적용될 수 있도록 한 부호이다.

$\boldsymbol{B}$ 행렬의 i 번째 행을 $\boldsymbol{B}[i]$라고 표시하기로 하자. BP 복호 알고리즘에서는 차수가 1인 행 $\boldsymbol{B}[i]$에서 '1'의 열 위치가 j라고 하면 $m_j = y_i$ 로 복호하고 j 번째 열에 있는 모든 '1'들을 '0'으로 바꿈과 동시에 '0'으로 바뀌는 행에 해당하는 부호 패킷 y_l을 y_i와 xor 연산을 하게 된다. 다시 차수가 1인 행을 찾아 위와 같은 복호 과정을 $\boldsymbol{B}$ 행렬의 원소가 모두 0이 될 때까지 반복하는데

$\boldsymbol{B}$ 행렬의 원소가 모두 0이 되면 복호가 성공한 것이고 모두 0이 되지 않았는데도 불구하고 차수가 1인 행이 없으면 복호 과정이 중단되어 실패하게 된다. 복호 과정이 중단되면 새로운 패킷을 수신한 후 B 행렬의 모든 원소가 '0'이 될 때까지 위의 복호과정을 반복한다[6].

OFG 알고리즘은 $\boldsymbol{B}$ 행렬에 대해 xoring 및 swapping 과정을 통해 삼각화를 한 후 후방 대입법을 적용하여 해 $\boldsymbol{m}$을 구하므로 GE 과정에 기반을 두고 있다고 할 수 있다. 하지만 GE 복호와는 달리 k 개의 부호 패킷이 도착할 때까지 기다리지 않고, 처음 도착하는 패킷부터 xoring 및 swapping 과정을 시작하여 행렬 $\boldsymbol{B}$의 삼각화(triangularization)를 진행한다. 여기에서 swapping 과정은 '1'의 밀도가 작아지도록 하는 역할을 한다[7].

3. Xoring을 이용한 BP 알고리즘

일반적으로 BP 복호 과정에서 k 개의 수신 패킷들을 수신하여 BP 복호를 수행하면 약 6-8% 정도(그림 1의 화살표)의 소스 패킷을 복호하고 리플(ripple)이 비게 되어 복호과정이 중단된다. BP 복호기는 차수-1의 수신 패킷을 기다리며 계속하여 패킷을 수신하다가 차수-1의 수신 패킷이 도착하면 BP 복호 과정을 재개하게 된다.

그림 1은 $c=0.1,\ \delta=0.01,\ k=1000$일 때 BP 복호기의 리플 추이를 나타낸 것으로 리플이 빌 때마다 차수-1의 패킷이 새롭게 도착할 때까지 허비되는 패킷들이 과다함을 볼 수 있다. 이러한 과정은 모든 소스 패킷들이 복호될 때까지 반복되다가 쌓여서 최종적으로 복호가 끝날 때에는 그림 1에서 볼 수 있는 바와 같이 차수-1 이상의 패킷은 물론 리플에 남아 사용되지 않고 버려지는 차수-1의 패킷도(그림 1의 경우 349개) 과다함을 볼 수 있다. 이러한 사실은 오버헤드 증가로 이어지게 된다. 본 논문에서는 앞 절에서 설명한 바와 같이 BP 복호기의 과다한 오버헤드를 줄이기 위한 복호 알고리즘을 제시하고자 한다.

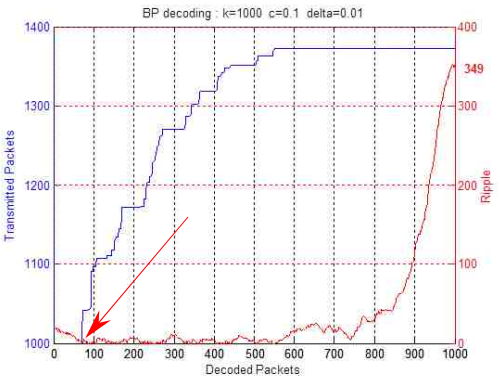

그림 .1 BP 복호기의 리플 추이($c=0.1,\ \delta=0.01,\ k=1000$)

Fig. 1. Ripple transition pattern of BP decoder ($c=0.1,\ \delta=0.01,\ k=1000$)

OFG 복호 알고리즘에서 사용하는 xoring 연산과 swapping 과정은 원래 행렬 $\boldsymbol{B}$를 삼각화(triangularization)하고 1의 밀도를 낮게 하기 위해 사용한다. 하지만 이들 두 기능은 행렬 $\boldsymbol{B}$를 구성하는 비트열 b_i의 차수를 감소시키는 중요한 역할을 하며 복호 과정이 끝날 때 까지 OFG 리플에 원소들을 충분히 공급해 주는 중요한 기능을 가지고 있다. 본 논문에서는 이러한 점에 착안하여 BP 복호 도중 리플이 비어 복호과정이 중단될 경우, 기존의 BP 복호 알고리즘에서와 같이 차수-1의 패킷이 도착할 때 까지 기다리지 않고 차수-2 이상의 패킷이 수신되어도 기존 패킷들과 xoring 및 swapping 연산을 하여 차수-1을 갖는 패킷을 적극적으로 탐색함으로서 복호 과정이 중단되지 않도록 한다. OFG 알고리즘과 같이 swapping을 통해 1의 밀도가 감소(sparse) 함으로서 리플이 빌 확률이 낮아진다. swapping은 단순히 비트열b_i와 b_j를 교환하는 것이므로 연산 복잡도는 증가하지 않으며 xoring 연산에 의해서만 복잡도가 증가한다. 본 논문에서 제안한 개선된 BP 복호 알고리즘을 요약하면 다음과 같고, 그림 2는 이를 적용한 예를 나타낸 것이다.

(1) k 개의 패킷을 수신한 후 기존의 BP 복호 알고리즘을 적용하여 복호한다.

(2) 리플이 비어 복호 과정이 중단되면 새로운 패킷 정보 b_j를 수신하고 이미 복호된 비트 위치를 0으로 바꾼다.

(3) 좌측으로부터 b_i와 b_j의 열이 1의 값을 가지면 $b_k = b_i \oplus b_j$ xoring 연산을 적용한다. Xoring 연산 시 $W(b_j) \le W(b_i)$ 이면 swapping을 수행한다. 단, $W(b_l)$은 b_l 벡터에 존재하는 1의 개수이다.

(4) Xoring 연산과 swapping 과정이 종료된 후 b_k가 전영 벡터가 아니면 $\boldsymbol{B}$ 행렬에 저장한다.

(5) 차수-1의 패킷을 탐색하여 BP 복호를 재개한다. 복호가 종료되지 않았는데도 차수-1의 패킷이 없으면 (2)~(5)의 과정을 반복한다.

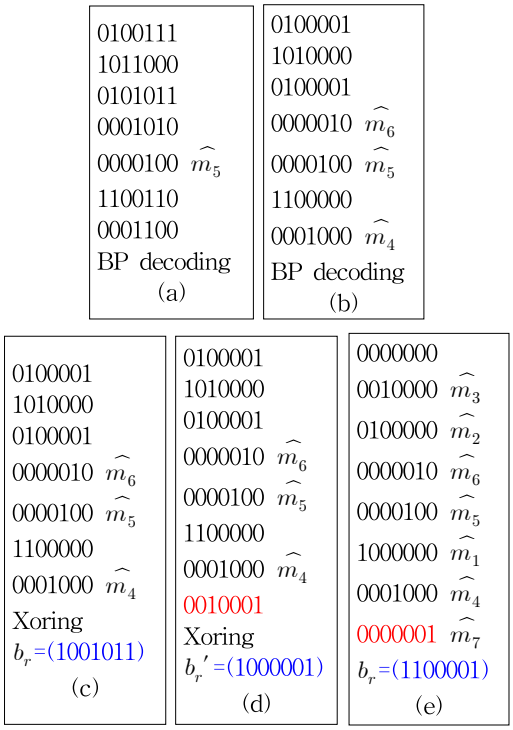

그림 2. Xoring을 적용한 BP 복호의 예

Fig. 2. An example of BP decoding with Xoring operation

그림 2-(a)에서 m_5가 리플에 저장되고 BP 복호가 진행되면 그림 2-(b)와 같이 m_4, m_6가 복호 된 후 중단된다. 새로운 패킷이 수신되어 정보 벡터 b_r을 그림 2-(c)와 같이 얻으면 차수가 4이므로 Xoring 연산 및 swapping을 적용한다. 이때 메시지 m_4, m_5 및 m_6 가 복호되어 있으므로 해당 위치를 0으로 바꾸면 b_r'을 얻게 되어 그림 2-(d)의 결과를 얻는다. 차수-1의 벡터가 없으므로 다시 새로운 패킷을 수신하고 그림 2-(e)와 같이 해당 정보 벡터 b_r을 얻어 Xoring 연산을 수행하여 나머지 메시지를 모두 복호한다.

본 논문에서 제안한 개선된 BP 복호 알고리즘은 OFG와 같이 삼각화(triangularization)가 필요하지 않다. 기존 BP 복호 알고리즘과 동일한 방법으로 복호를 시작하고 차수-1 패킷의 부족으로 복호 과정이 중단되면 OFG 알고리즘에서 사용하는 Xoring 연산과 swapping을 이용해 차수-1의 패킷을 능동적으로 발생시켜 복호를 완성하게 된다. 삼각화 과정이 없으므로 후방 대입법에 의한 가우시안 제거 과정도 필요하지 않으며 기존의 BP 복호에 비해 연산량이 증가하는 부분은 xoring 과정 밖에 없다. Xoring 및 swapping 과정을 통해 $\boldsymbol{B}$의 1 값의 밀도는 점차 낮아지고 차수-1의 벡터가 존재할 확률은 커져 리플은 복호가 완료될 때까지 비워지지 않는다.

4. 시뮬레이션 결과 및 분석

본 논문에서 제시한 개선된 BP 복호 알고리즘의 성능을 고찰하기 위해 다음과 같은 모의실험을 수행하였다. 부호 길이는 $k = 100 \sim 1000$으로 하였으며 오버헤드와 복호 연산량에 대해 성능을 측정하였다. δ 값은 0.01로 정하였으며 리플의 크기에 영향을 주는 c 값은 0.1로 하여 기존의 BP 알고리즘과 제안된 알고리즘을 대상으로 실험하였다.

[표 1] k에 따른 오버헤드(c=0.1, $\delta=0.01$)
[Table 1] Overhead vs k(c=0.1, $\delta=0.01$)

Decoding Scheme / Code Length (k)	BP	Improved BP
100	0.657240	0.048820
200	0.548430	0.024890
300	0.479947	0.015380
400	0.447185	0.015545
500	0.427604	0.012492
600	0.391673	0.008087
700	0.381020	0.005414
800	0.364150	0.005102
900	0.353740	0.004460
1000	0.342586	0.003328

[표 1]은 $c=0.1$이고 $\delta=0.01$일 때 k 값에 따른 오버헤드 비중을 기존의 BP 복호 알고리즘과 본 논문에서 제안한 BP 알고리즘에 대해 시뮬레이션을 수행한 결과를 나타낸 것이다. 총 수신된 패킷 수를 n이라고 하고 LT 부호 길이를 k라고 하면 오버헤드는 $\epsilon = n/k-1$으로 표현할 수 있다. 기존의 BP 복호 알고리즘은 최소 약 30% 이상의 오버헤드가 필요하며 k값이 감소할수록 오버헤드 비중은 점차 증가하여 60%에 육박함을 볼 수 있다. 이에 비해 제안된 알고리즘의 경우 $k=1000$일 경우 약 0.33%에 불과하며 $k=50$일 때에도 5% 미만의 오버헤드를 나타낸다.

[표 2]는 $c=0.1$이고 $\delta=0.01$일 때 k 값에 따른 연산 복잡도(computational complexity)를 나타낸 것이다. 삼각화 과정이 없으므로 후방 대입법에 의한 가우시안 제거 과정도 필요하지 않으며 기존의 BP 복호에 비해 연산량이 증가하는 부분은 xoring 과정 밖에 없다. [표 2]에서 기존의 BP 복호 알고리즘과 제안된 알고리즘의 복잡도는 한 자리수 정도의 복잡도 밖에 차이를 보이지 않으며 k 값이 감소할수록 복잡도는 거의 동일함을 알 수 있다. 따라서 k 값이 크지 않은 경우 제안된 알고리즘의 성능은 기존 BP 알고리즘에 비해 오버헤드와 복잡도 면에서 크게 우수함을 알 수 있다.

[표 2] k 에 따른 복호 복잡도 (c=0.1, $\delta=0.01$)
[Fig 2] Complexity vs k (c=0.1, $\delta=0.01$)

Decoding Scheme / Code Length (k)	BP	Improved BP
100	1371.48	1441.382
200	2603.16	4101.420
300	4199.36	8568.430
400	5694.50	14156.692
500	7360.952	21521.184
600	8778.874	29311.654
700	10305.786	39039.494
800	10965.366	49887.502
900	13836.662	61989.732
1000	16640.720	74578.732

5. 결 론

본 논문에서는 연산 복잡도를 증가시키지 않으면서도 BP 복호의 단점인 오버헤드를 줄이기 위해 OFG 복호 기법의 Xoring을 이용한 개선된 BP 복호 알고리즘을 제안하였다. 시뮬레이션 결과 기존의 BP 복호 알고리즘은 최소 약 30% 이상의 오버헤드가 필요하며 k값이 감소할수록 오버헤드 비중은 점차 증가하여 60%에 육박함을 볼 수 있었다. 이에 비해 제안된 알고리즘은 $k=1000$일 경우 약 0.33%에 불과하였으며 $k=50$일 때에도 5% 미만의 오버헤드를 나타내었다. 연산 복잡도의 경우 기존의 BP 복호 알고리즘과 제안된 알고리즘의 복잡도는 한 자리수 정도의 복잡도 밖에 차이를 보이지 않았으며 k 값이 감소할수록 복잡도는 거의 동일한 수준으로 감소함을 알 수 있었다.

감사의 글

본 논문은 남서울대학교 2012 교내연구과제로 수행되었음.

References

[1] M. Luby, "LT codes," Proceedings 43rd Annual IEEE Symposium on Foundations of Computer Science, pp. 271~280, Vancouver, Canada, Nov. 2002.

[2] D.J.C. MacKay, "Fountain Codes," IEE Proceedings -Communications, Vol. 152, No. 6, December 2005, pp. 1062-1068.

[3] Valerio Bilglio, Macro Grangetto, Rossano Gaeta, Matteo Sereno, "On the fly Gaussian Elimination for LT codes," IEEE Communication Letters, Vol. 13, No. 12, pp. 953~955, Dec. 2009.

[4] Hoyoung Cheong, " An Efficient Decoding for Short Block Length LT Codes," Journal of Convergence Information Technology, JCIT, Vol. 7, No. 9., 2012. 5, pp. 76-82.

[5] Saejoon Kim, Karam Ko, and Sae-Young Chung, "Incremental Gaussian Elimination Decoding of Raptor Codes over BEC," IEEE Communication Letters, Vol. 12, No. 14, pp. 307~309, April 2008.

[6] Hoyoung Cheong, "An Efficient Fountain Code for Vehicular Communication Channel," Proceedings of 2012 KIIECT Conference, Vol.5, No. 1, May, 2012.

[7] A. Shokrollahi : 'Raptor Codes', IEEE Transactions on Information Theory, Vol. 52, No. 6, June 2006, pp. 2551 - 2567.

저자약력

정 호 영(Ho-Young Cheong) [종신회원]

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

<관심분야> 채널부호, 양자 암호