https://doi.org/10.7840/kics.2016.41.12.1768

상대 차족을 이용한 복수 반복 차수 분산 저장 부호

박 호 성◆, 김 철 성°

Distributed Storage Codes with Multiple Replication Degrees Using Relative Difference Families

Hosung Park◆, Cheol-Sung Kim°

요 약

본 논문에서 상대 차족을 이용하여 분산 저장 부호의 한 종류인 분할 반복 부호를 설계하는 방법을 제안한다. 제안하는 부호는 각 데이터 심볼마다 다른 반복 차수를 지원할 수 있다. 모의실험을 통하여 제안하는 부호가 랜덤 분할 반복 부호보다 더 많은 양의 데이터를 저장할 수 있음을 보인다.

Key Words : Distributed storage code, fractional repetition code, relative difference family, replication degree

ABSTRACT

In this paper, we propose a construction method of fractional repetition codes, a class of distributed storage codes, using relative difference families. The proposed codes can support multiple replication degrees for data symbols. It is shown via simulation that the proposed codes store more data than the random fractional repetition codes.

Ⅰ. 서 론

향후 데이터 저장 요구량이 급격히 증가할 것으로 예상됨에 따라 분산 저장 시스템의 효율성 및 신뢰성이 매우 중요해지고 있다. 이를 위해 부분접속 복구 부호(locally repairable codes) [1]와 재생성 부호(regeneration code) [2]가 최근 제안되었다. 하나의 노드가 손실되었을 때 부분접속 복구 부호는 최소한의 다른 노드들에 접속함으로써 그 노드를 복구하는 것을 목표로 하는 반면, 재생성 부호는 네트워크의 대역폭을 최소로 사용하여 복구하는 것을 목표로 한다. 재생성 부호에서 저장량과 복구 대역폭 간에 균형 관계가 존재하며 [2], 그 양 극단이 각각 최소 저장 재생성 부호와 최소 대역폭 재생성 부호에 해당한다.

부분 반복 부호 [3]는 저장하고자 하는 데이터를 여러 블록들로 분할하고, 각각을 여러 번 복제한 후, 이것들을 노드들에 일정한 규칙에 근거하여 분산시킨다. 부분 반복 부호는 최소 대역폭 재생성 부호의 변형으로서 복구를 위한 네트워크 대역폭을 최소한으로 이용할 뿐 아니라 디스크의 입출력 및 연산을 최소로 하는 특징을 가지고 있다. 현재까지 그래프 이론, 디자인 이론 등을 이용하여 다양한 부분 반복 부호들이 제안되었고, 각 설계 방법마다 부호의 파라미터들이 다르므로 보다 다양한 부호 파라미터를 위해서 새로운 설계 방법들이 필요하다.

본 논문에서는 상대 차족을 이용하여 새로운 분할 반복 부호들을 설계하는 일반적인 방법을 제시한 후, 실제 설계된 다양한 상대 차족들을 기반으로 [4] 제안하는 부호의 설계 가능한 파라미터들을 제시한다. 그 결과 제안하는 분할 반복 부호들은 노드당 데이터 저장량은 일정하면서 파일의 각 부분마다 다양한 반복 차수를 가질 수 있도록 한다. 제안하는 부호들과 [5]에서 제안한 랜덤 분할 반복 부호를 모의실험으로 비교하여 제안하는 부호로 더 많은 데이터를 저장할 수 있음을 보인다.

Ⅱ. 분할 반복 부호

본 논문에서 데이터의 각 심볼들은 F_q의 원소이고, 파일은 θ개의 심볼로 이루어지며, 데이터를 저장하기 위한 노드의 수는 n이라고 가정하자. 정수를 원소로 가지는 멀티셋 $A=\{\alpha_0,\ldots,\alpha_{n-1}\}$와 $R=\{\rho_0,\ldots,\rho_{\theta-1}\}$에 대해 (n,A,R) 분할 반복 부호 C는 다음의 조건을 만족시키는 $\{0,1,\ldots,\theta-1\}$의 n개의 부분집합 $N_0,N_1,\ldots,N_{n-1}$으로 정의된다.

※ 본 연구는 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업이며 (No. NRF-2015R1D1A1A01060941), 또한 2015년도 전남대학교 학술연구비 지원에 의하여 연구되었음
◆ First Author : School of Electronics and Computer Engineering, Chonnam National University, hpark1@jnu.ac.kr, 종신회원
° Corresponding Author : School of Electronics and Computer Engineering, Chonnam National University, chskim@jnu.ac.kr, 정회원
논문번호 : KICS2016-10-313, Received October 16, 2016; Revised November 8, 2016; Accepted November 14, 2016

- $\sum_{i=0}^{n-1}\alpha_i = \sum_{j=0}^{\theta-1}\rho_j$;
- $|N_i| = \alpha_i,\ i = 0,...,n-1$;
- 파일의 심볼 j는 C의 부분집합들에 정확히 ρ_j 번 포함된다.

위와 같이 정의된 분할 반복 부호 C에 의해 노드 i는 N_i에 있는 심볼들을 저장한다. 파라미터 ρ_j는 각 심볼에 대한 반복 차수라고 불린다. 분할 반복 부호 C는 접속 행렬 $I(C)$에 의해 표현되기도 하는데, 이는 $n \times \theta$의 크기를 가지고 그 (i,j)원소는 N_i가 심볼 j를 포함하면 1, 그렇지 않으면 0이 된다. 모든 α_i가 α로 일정하고, ρ_j 역시 ρ로 일정할 때 C는 (n,α,ρ) 정규 분할 반복 부호라고 불리고, 그 접속행렬은 행가중치와 열가중치가 각각 일정한 정규 행렬이 된다. 정규 분할 반복 부호가 아니면 비정규 분할 반복 부호라 부른다.

분할 반복 부호를 사용한 분산 저장 장치에서 하나의 노드 i가 손실되었을 때 다른 α_i개의 노드들로부터 심볼 하나씩을 다운로드 받으면 그 노드는 복구될 수 있다. 그리고 사용자가 임의의 k개의 노드에 접속해서 다운로드 받을 수 있는 최소의 심볼 수 $M(k)$는 다음과 같이 표현된다.

$$M(k) = \min_{\substack{I \subset \{0,...,n-1\} \\ |I| = k}} |\cup_{i \in I} N_i|$$

정규 분할 반복 부호에 대하여 $M(k)$에 대한 상계가 [3]에 알려져 있다.

Ⅲ. 상대 차족을 이용한 분할 반복 부호 설계

본 논문에서는 [4]에서 제안한 다양한 블록 사이즈를 가지는 상대 차족에 기반하여 새로운 (n,α,R) 비정규 분할 반복 부호를 제안하고자 한다. 제안하는 부호는 모든 노드들은 동일한 양의 심볼들을 저장하는 반면, 저장하고자 하는 파일의 각 심볼들은 중요도에 따라 다른 반복 차수를 가질 수 있게 하는 특징이 있다. 이는 네트워크의 로드 균형을 유지하면서도 파일의 특성을 감안한 효율적인 저장 방법을 의미한다.

본 논문에서 표기법의 편의를 위해 (n,α,R) 분할 반복 부호에서 R을 θ개의 심볼에 대한 반복 차수를 모두 나열하는 것이 아니라 각 반복 차수에 대한 빈도수를 표시하도록 한다. 예를 들어 $\{2,2,2,3,3,4\}$는 $\{2^3,3^2,4^1\}$로 표시한다.

연산 $\cdot$ 로 정의되는 군 $G = \{e_0,...,e_{g-1}\}$에 대하여 그 정규 부분군 H가 h개의 원소를 가진다고 하자. 그리고 양수들을 원소로 가지는 집합 K를 가정하자. $(g,h,K,1)$-상대 차족은 G의 부분 집합들(블록이라고 부르기로 한다)의 모임으로 정의되며, 각 블록의 크기는 K의 원소이며 각 블록 내의 원소들의 차를 모든 블록에 대하여 모아 놓을 때 $G \backslash H$의 모든 원소들이 정확히 1번 나오는 성질을 만족한다. 그리고 군 $G = \{e_0,...,e_{g-1}\}$의 임의의 원소 e에 대하여 다음과 같은 연산자를 정의한다.

$$< e > = i \text{ if } e = e_i \text{ for } 0 \le i \le g-1$$

$(g,h,K,1)$-상대 차족이 t개의 블록 $B_0,...,B_{t-1}$로 이루어져 있고 각 블록의 크기를 $k_j \in K$ $(j = 0,...,t-1)$라 하고, j번째 블록을 $B_j = \{b_{j,0},...,b_{j,k_j-1}\}$으로 표시하자. 제안하는 분할 반복 부호는 노드 i가 $(0 \le i \le g-1)$ $\bigcup_{j=0}^{t-1}\{jg + < e_i \cdot b_{j,0} >, ..., jg + < e_i \cdot b_{j,k_{j-1}} >\}$에 해당하는 심볼들을 저장하도록 한다. 이렇게 정의된 분할 반복 부호는 파라미터 $n = g,\ \alpha = \sum_{i=0}^{t-1} k_i,\ \theta = gt,$ $\rho_j = k_{\lfloor j/g \rfloor}\ (j = 0,...,\theta-1)$을 갖는다. [4]에서 제안한 상대 차족을 이용하면 표 1과 같이 다양한 파라미터

표 1. 상대 차족에 따른 제안하는 부호의 파라미터
Table 1. Parameters of the proposed codes and the corresponding relative difference families

Relative difference families	Parameters of the proposed FR codes
$(19p, 19, \{3,4,5\}, 1)$ $p \ge 5$ is an odd prime	$n = 19p,\ \alpha = 6(p-1),$ $R = \{3^{19p(p-1)/2}, 4^{19p(p-1)/2}, 5^{19p(p-1)/2}\}$
$(22p, 22, \{3,4,5\}, 1)$ $p \ge 5$ is an odd prime	$n = 22p,\ \alpha = 15(p-1)/2,$ $R = \{3^{22p(p-1)}, 4^{11p(p-1)}, 5^{11p(p-1)}\}$
$(28p, 28, \{3,4,5\}, 1)$ $p \ge 5$ is an odd prime	$n = 28p,\ \alpha = 19(p-1)/2,$ $R = \{3^{28p(p-1)}, 4^{28p(p-1)}, 5^{14p(p-1)}\}$
$(25p, 25, \{3,4,5\}, 1)$ $p \ge 5$ is an odd prime	$n = 25p,\ \alpha = 9(p-1),$ $R = \{3^{75p(p-1)/2}, 4^{25p(p-1)/2}, 5^{25p(p-1)/2}\}$
$(24p, 24, \{3,4,6\}, 1)$ $p \ge 7$ is an odd prime	$n = 24p,\ \alpha = 13(p-1)/2,$ $R = \{3^{12p(p-1)}, 4^{12p(p-1)}, 6^{12p(p-1)}\}$
$(28p, 28, \{3,5,6\}, 1)$ $p \ge 7$ is an odd prime	$n = 28p,\ \alpha = 7(p-1),$ $R = \{3^{14p(p-1)}, 5^{14p(p-1)}, 6^{14p(p-1)}\}$
$(34p, 34, \{3,4,5,6\}, 1)$ $p \ge 7$ is an odd prime	$n = 34p,\ \alpha = 9(p-1),$ $R = \{3^{17p(p-1)}, 4^{17p(p-1)}, 5^{17p(p-1)}, 6^{17p(p-1)}\}$

터들을 가지는 분할 반복 부호를 생성할 수 있다.

한편, 위에서 제안한 분할 반복 부호의 설계 방법에서 t개의 모든 블록들을 이용하는 것이 아니라 $t' < t$개의 블록만을 이용하여 다양한 데이터 심볼 수를 지원하는 분할 반복 부호를 설계할 수 있다. 이 때도 모든 노드에 대하여 α는 일정한 값을 가지며 $\theta = gt'$이 된다. 이렇게 생성된 제안하는 부호들은 상대 차족의 정의에 의해 임의의 두 개의 노드들을 선택했을 때 h/g의 비율만큼은 공통 심볼을 가지지 않고 나머지는 1개의 공통된 심볼을 가지므로 큰 $M(k)$를 가지게 된다. 이는 일반 차족에 의한 설계보다 특성이 더 좋다고 할 수 있다.

Ⅳ. 모의실험

제안하는 분할 반복 부호의 성능을 검증하기 위해서 k개의 노드에 접속하여 데이터를 읽었을 때 보장할 수 있는 심볼의 수를 모의실험으로 구하고, 이를 [5]에서 제안한 랜덤 분할 반복 부호와 비교하도록 한다. [5]에서는 대규모 분산 저장 장치를 가정하여 비교적 큰 부호들을 설계하였는데, 여기서도 제안하는 부호로서 $(95,19,\{3,4,5\},1)$-상대 차족으로부터 $(95,24,\{3^{190},4^{190},5^{190}\})$ 부호를 생성하였고, 이를 동일한 파라미터의 랜덤 분할 반복 부호와 비교한다. 그림 1에서 볼 수 있듯이 동일한 k에 대하여 제안하는 부호가 랜덤 부호보다 읽을 수 있는 데이터 심볼의 개수가 더 큰 것을 볼 수 있다.

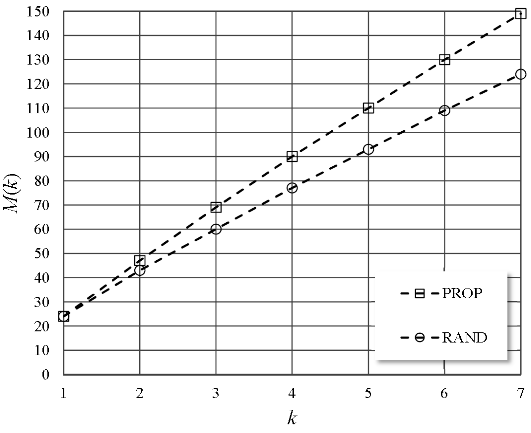

그림 1. 제안하는 $(95,24,\{3^{190},4^{190},5^{190}\})$ 분할 반복 부호와 랜덤 분할 반복 부호의 k에 따른 심볼 수 비교
Fig. 1. Comparison of the number of symbols between the proposed $(95,24,\{3^{190},4^{190},5^{190}\})$ FR code and the random FR code with respect to k

Ⅴ. 결 론

본 논문에서 $(g,h,K,1)$-상대 차족으로부터 (n,α,R) 분할 반복 부호를 설계하는 방법을 제시하였다. 다양한 상대 차족의 설계 방법이 알려져 있기 때문에 제안하는 부호들을 역시 다양하게 설계할 수 있다. 모의설험을 통하여 제안하는 부호가 랜덤 분할 반복 부호보다 더 많은 심볼들을 저장할 수 있음을 보였다.

References

[1] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5843-5855, Oct. 2014.

[2] J. S. Park, J.-H. Kim, K.-H. Park, and H.-Y. Song, "Average repair read cost of linear repairable code ensembles," *J. KICS*, vol. 39, no. 11, pp. 723-729, Nov. 2014.

[3] S. E. Rouayheb and K. Ramchandran, "FR codes for repair in distributed storage systems," in *Proc. 48th Annu. Allerton Conf. Contr., Comput., Commun.*, pp. 1510-1517, Urbana-Champaign, IL, USA, Sept. 2010,

[4] M. Buratti, Y. Wei, D. Wu, P. Fan, and M. Cheng, "Relative difference families with variable block sizes and their related OOCs," *IEEE Trans. Inf. Theory*, vol. 57, no. 11, pp. 7489-7497, Nov. 2011.

[5] S. Pawar, N. Noorshams, S. E. Rouayheb, and K. Ramchandran, "DRESS codes for the storage cloud: Simple randomized constructions," in *Proc. IEEE Symp. Inf. Theory*, pp. 2338-2342, St. Petersburg, Russia, Aug. 2011.