IR-심플렉스 부호 설계 및 부분접속수 분석

김정현, 남미영, 박기현, 송홍엽
연세대학교, 채널 부호 및 암호 연구실

2015 / 01 / 22

2015 한국통신학회 동계종합학술발표회

Abstract

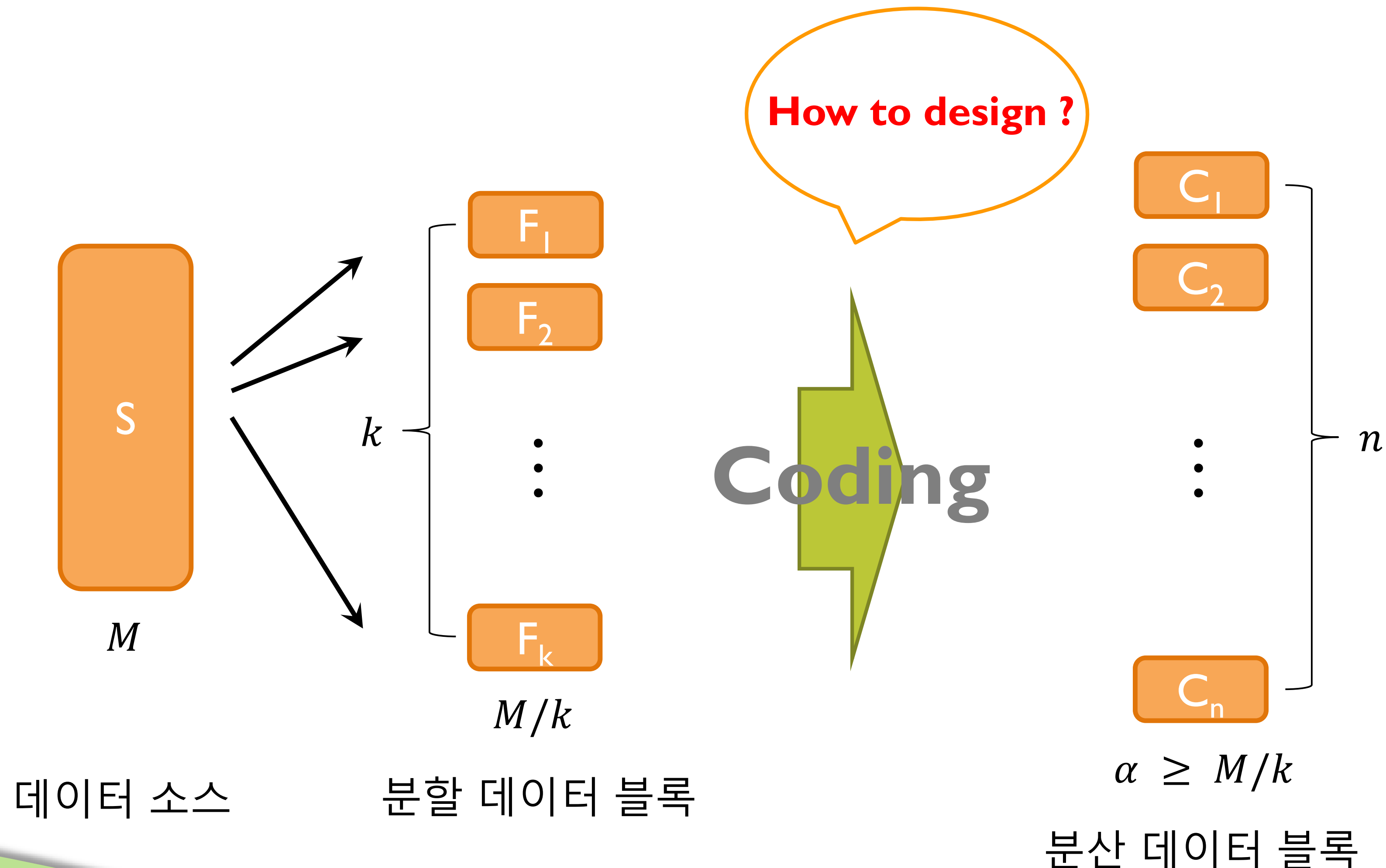
- 필요성
  - 분산 저장 시스템에서 빈번하게 노드(데이터) 소실 발생

- 목표
  - 데이터 소실을 효율적으로 복구하기 위한 부호 설계

- 접근
  - 낮은 복잡도로 데이터 복구 시 필요한 접속 노드 수를 최소화하는 부호에 대한 분석

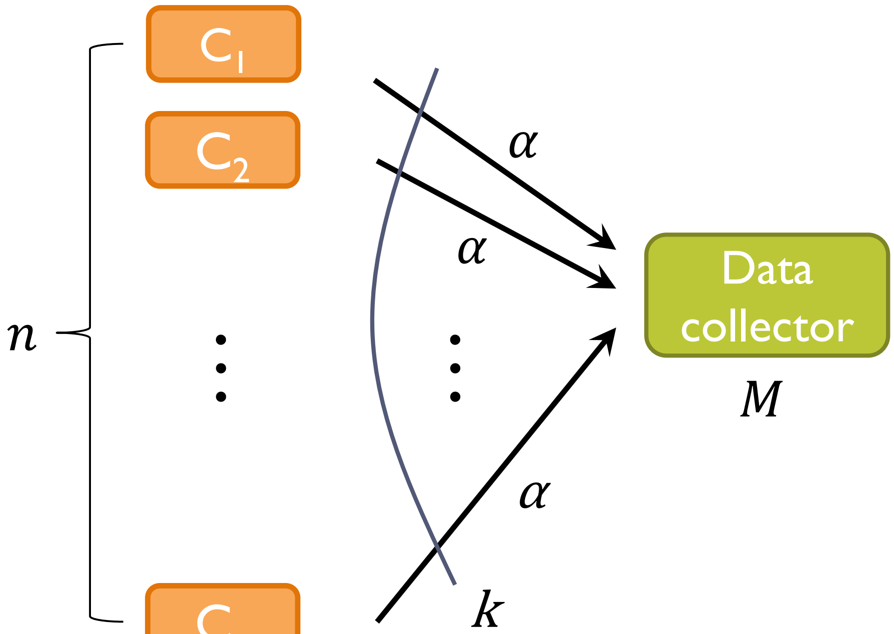
Distributed Storage Systems

- 데이터 소실을 극복하기 위해 부호화 사용



Distributed Storage Systems

- 데이터 수집 과정



n : 저장 노드의 수

k : 데이터 수집을 위한 저장 노드의 수

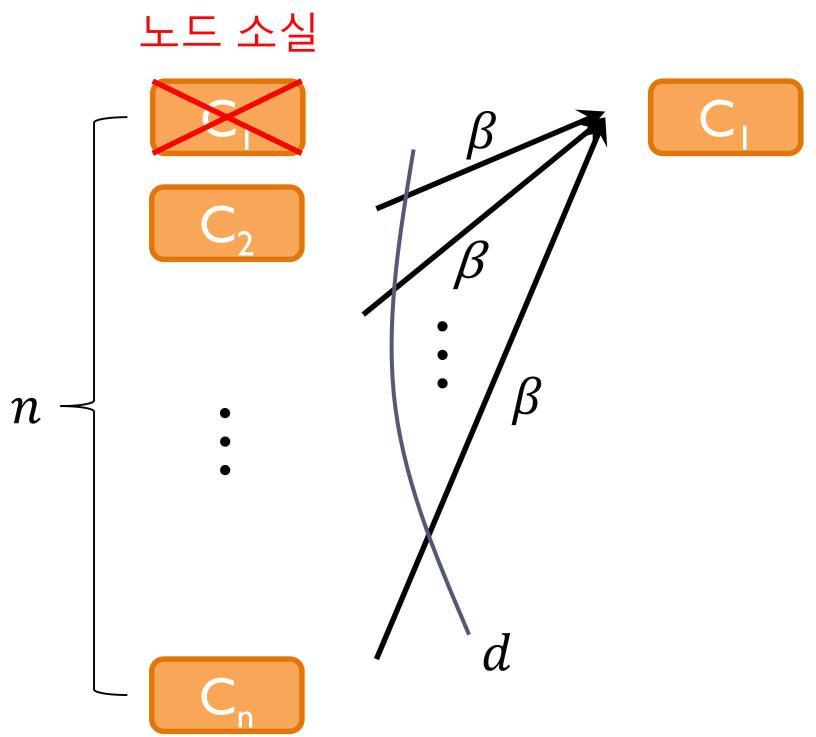
α : 저장 노드의 용량

M : 데이터 소스의 크기

분산 데이터 블록　　　　사용자

Distributed Storage Systems

- 데이터 복구 과정



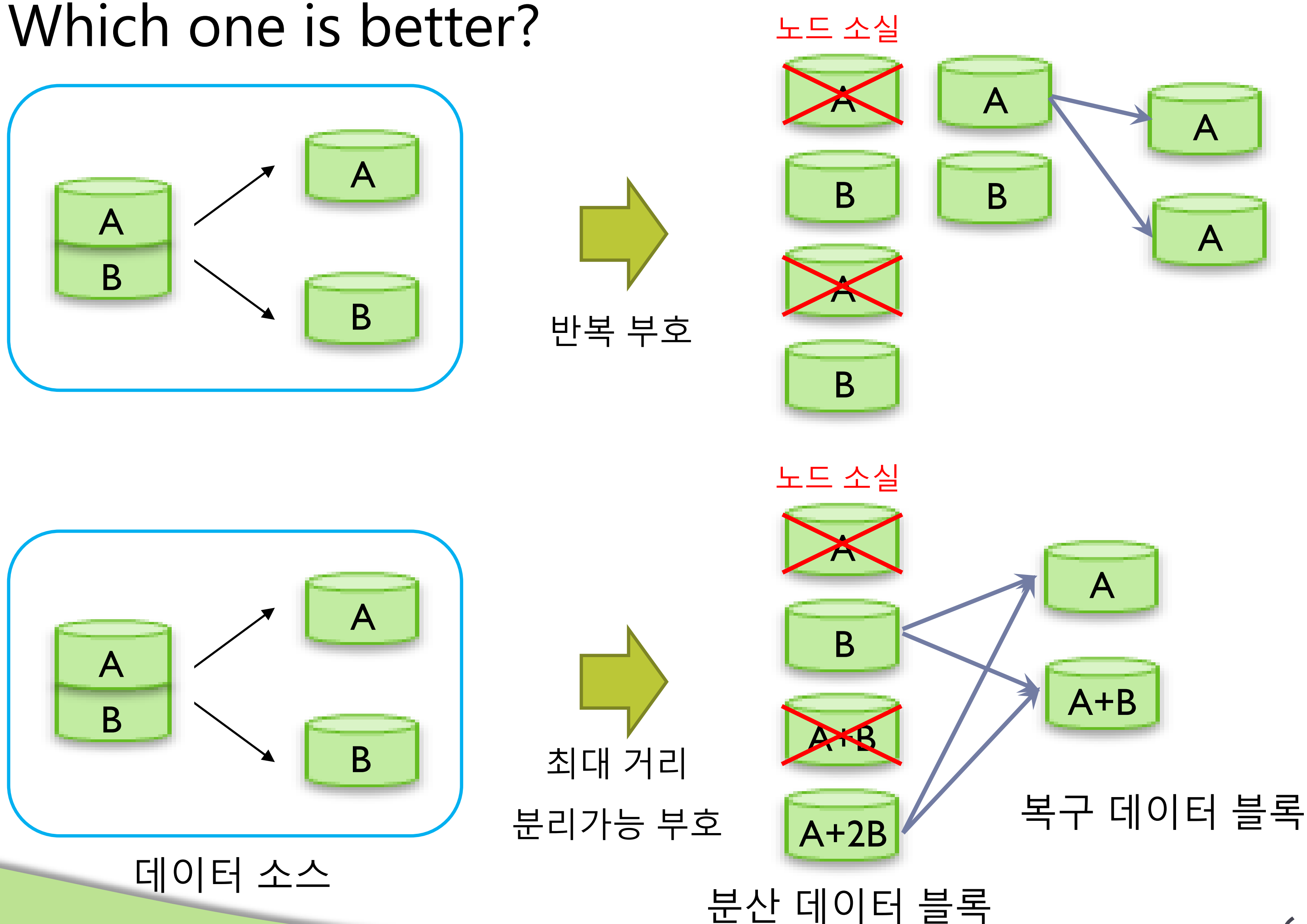
d : 데이터 복구를 위한 저장 노드의 수

β : 데이터 복구 시 노드들로부터 다운받는 데이터양

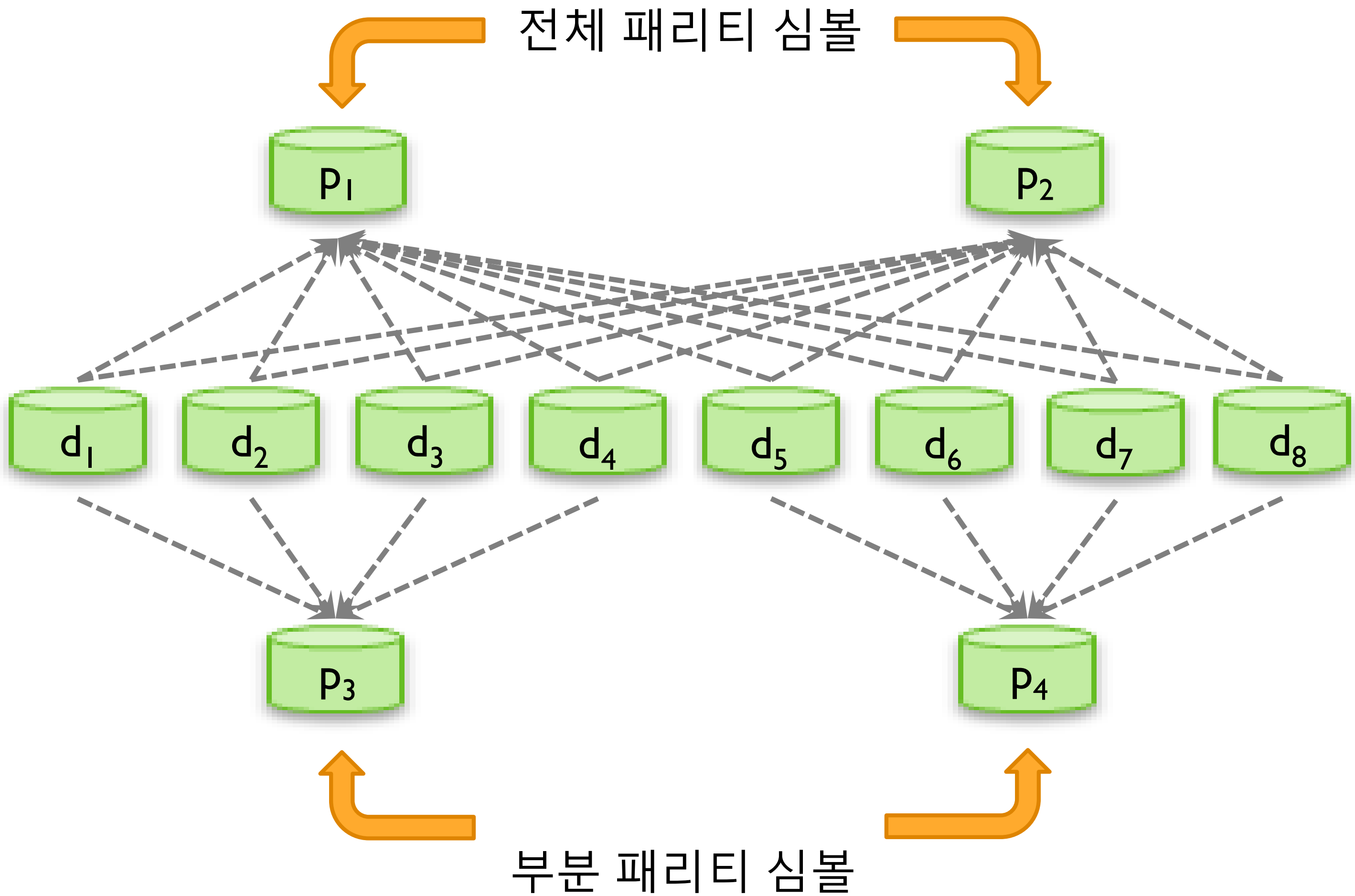
분산 데이터 블록 복구 데이터 블록

Distributed Storage Codes

- Which one is better?



Locally Repairable Codes



Simplex Codes and IR-simplex Codes

- Definition 1 [8]

임의의 양의 정수 k에 대해, $n = 2^k - 1$ 그리고 G를 각 열들이 $\mathbb{F}_2^k$ 상에서 영벡터가 아닌 서로 다른 벡터로 정의되는 $k \times n$ 행렬이라 하자. 그리고 부호 $\mathcal{C}$는 G를 생성행렬로 갖는 $[n, k, d]_2$ 부호라 하자. 이러한 부호 $\mathcal{C}$를 최소 거리 $d = 2^{k-1}$을 갖는 심플렉스 부호라 부른다.

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$[7,3,4]_2$ 심플렉스 부호의 생성행렬

[8] W.C. Huffman, V. Pless, "Fundamentals of Error Correcting Codes," University Press, Cambridge, 2003.

Simplex Codes and IR-simplex Codes

- Definition 2 [6]

G를 심플렉스 부호의 생성행렬에 항등행렬을 추가한 행렬이라고 하자. 즉, $G = (I|G_S)$이다. 여기서 I는 항등행렬이고, G_S는 심플렉스 부호의 생성행렬이다. 부호 $\mathcal{C}$는 G를 생성행렬로 갖는 $[n,k,d]_2$ 부호라 하자. 이러한 부호 $\mathcal{C}$를 길이 $n = 2^k - 1 + k$ 그리고 최소거리 $d = 2^{k-1} + 1$을 갖는 IR-심플렉스 부호라 부른다.

$$G = \left(\begin{array}{ccc|ccc|cccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right)$$

$[10,3,5]_2$ IR-심플렉스 부호의 생성행렬

[6] J.-H. Kim, and H.-Y. Song, "Simple construction of [2^k-1+k, k, 2^{k-1}+1] code attaining the Griesmer bound," 9th Joint Conference on Communications & Information, April 1999.

Repair Sets and Locality

- Definition 3

$\mathcal{C}$는 $[n,k,d]_2$ 부호라 하자. 임의의 정수 $i \in \{1,2,\ldots,n\}$에 대해, $\mathcal{C}$의 부호화된 심볼 Y_i가 다음 조건을 만족한다고 하자.

$$Y_i = \sum_{j \in R(i), R(i) \subseteq \{1,2,\ldots,n\} \setminus \{i\}} Y_j$$

그러면 이러한 $R(i)$를 Y_i의 복구집합이라고 한다. 또한 하나의 심볼에 대해 이러한 복구집합은 여러 개 일 수 있다.
가장 작은 크기를 갖는 복구집합의 원소의 수를 Y_i의 부분접속수라 한다.
모든 심볼의 부분접속수 중 가장 큰 값을 부호 $\mathcal{C}$의 부분접속수라 한다.

(Example)

만약, $Y_1 = Y_2 + Y_4$이면 Y_1의 복구집합 $R(1)$은 $\{2,4\}$이다.

Locality of Codes

Remark 1 [9]

$\mathcal{C}$는 $[n,k,d]_2$ 심플렉스 부호라 하자. 그러면 $\mathcal{C}$의 부분접속수는 2이다.

Theorem 1

$\mathcal{C}$는 $[n,k,d]_2$ IR-심플렉스 부호라 하자. 그러면 $\mathcal{C}$의 부분접속수는 2이다.

Proof:
IR-심플렉스 부호의 생성행렬의 구조로부터 증명할 수 있다.

[9] M. Kuijper and D. Napp. (2014). "Erasure codes with simplex locality," [Online]. Available: http://arxiv.org/abs/1403.2779.

Repair Sets of Codes

부호	심볼별 최소 크기를 갖는 복구집합	평균 부분 접속수
$[9,3,\mathbf{3}]_2$ 3회 반복 부호	$R(1)=\{4\},\{7\},\quad R(2)=\{5\},\{8\},\quad R(3)=\{6\},\{9\},$ $R(4)=\{1\},\{7\},\quad R(5)=\{2\},\{8\},\quad R(6)=\{3\},\{9\},$ $R(7)=\{1\},\{4\},\quad R(8)=\{2\},\{5\},\quad R(9)=\{3\},\{6\}$	**1**
$[7,3,\mathbf{4}]_2$ 심플렉스 부호	$R(1)=\{2,4\},\{3,5\},\{6,7\},\ R(2)=\{1,4\},\{3,6\},\{5,7\},$ $R(3)=\{1,5\},\{2,6\},\{4,7\},\ R(4)=\{1,2\},\{3,7\},\{5,6\},$ $R(5)=\{1,3\},\{2,7\},\{4,6\},\ R(6)=\{1,7\},\{2,3\},\{4,5\},$ $R(7)=\{1,6\},\{2,5\},\{3,4\}$	**2**
$[10,3,\mathbf{5}]_2$ IR-심플렉스 부호	$R(1)=\{4\},\qquad R(2)=\{5\},\qquad R(3)=\{6\},$ $R(4)=\{1\},\qquad R(5)=\{2\},\qquad R(6)=\{3\},$ $R(7)=\{1,2\},\{1,5\},\{2,4\},\{3,10\},\{4,5\},\{6,10\},\{8,9\},$ $R(8)=\{1,3\},\{1,6\},\{2,10\},\{3,4\},\{4,6\},\{5,10\},\{7,9\},$ $R(9)=\{1,10\},\{2,3\},\{2,6\},\{3,6\},\{4,10\},\{5,6\},\{7,8\},$ $R(10)=\{1,9\},\{2,8\},\{3,7\},\{4,9\},\{5,8\},\{6,7\}$	**1.4**

Conclusions

- IR-심플렉스 부호는 평균 부분접속수 측면에서 심플렉스 부호보다 우수함.
- IR-심플렉스 부호는 최소거리 측면에서 반복 부호 및 심플렉스 부호보다 우수함.
- 3회 반복 부호를 사용하는 하둡(Hadoop)과 같은 시스템에 용이하게 구현 가능함.
- 향후 연구 주제로 부호율의 손실을 최소화하면서 위 장점을 보장하는 새로운 부호 설계가 필요함.