

# 분산 부호화 캐싱 시스템에 대한 최적의 파일 저장 기법

이원석\*, 임승찬°, 박현철\*

## Optimal File Placement for Distributed Coded Caching Systems

Wonseok Lee\*, Seung-Chan Lim°, Hyuncheol Park\*

### 요약

본 논문에서는 채널 부호를 적용한 분산 캐싱 시스템에 대하여 최적의 파일 저장 기법을 연구한다. 기존 연구에서 고려하였던 MDS (maximum distance separable) 부호가 적용된 무선 캐싱 시스템을 일반화하여 상대적으로 더욱 실용적인 파운틴(fountain) 류 부호의 적용을 고려한다. 채널 부호의 복호 실패 확률을 반영하여 시스템의 서비스 실패 확률을 정의하고 이에 대한 하한 값을 분석한다. 또한, 분석된 하한 값에 기반하여 시스템의 서비스 실패 확률을 최소화하는 파일 분산 저장 알고리즘을 제안하고 다양한 시스템 파라미터에 대한 모의 실험을 수행한다. 제안된 파일 저장 기법은 기존 저장 기법의 서비스 실패 확률을 크게 개선 가능할 뿐만 아니라 분석된 하한 값에 매우 근접한 성능을 가짐을 모의 실험을 통하여 검증하였다.

**Key Words** : Wireless caching, distributed coded caching, optimal file placement, service failure probability, fountain codes

### ABSTRACT

In this paper, we study the optimal file placement strategy for distributed coded caching systems. By applying practical channels codes, such as fountain codes, we generalize the existing maximum distance separable (MDS) coded caching systems. Considering the decoding failure probability of the channels codes, we define the service failure probability of the systems, and analyze the lower bound of it. Based on the analyzed lower bound, we propose the file placement algorithm that can minimize the service failure probability. The simulation results show that the proposed algorithm can significantly improve the service failure probability of the existing methods, and the performance closely approaches to the lower bound of the systems.

### I. 서론

최근 스마트 기기의 활성화 및 대용량 멀티미디어 서비스 요구의 증가로 인하여 모바일 데이터 트래픽의 수요가 폭발적으로 증가하고 있다<sup>[1]</sup>. 급증하는 대용량 멀티미디어의 수요에 효과적으로 대응하기 위하

여 기존에는 주파수 효율 향상, 밀리미터파 대역의 추가적인 사용, 주파수의 효율적인 재사용 등의 다양한 방법으로 데이터 전송량을 증가시키려고 노력했으나 상대적으로 더 빠른 모바일 데이터 트래픽의 증가 속도에 대응하는 데 한계가 발생하고 있다. 이러한 한계를 극복하기 위해서 최근에는 저장 장치를 통신자원

※ 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2017-0-01973, 5G초고주파 기반 고속이동체 환경에서의 채널특성 및 성능평가 국제 공동연구)

• First Author : Hyundai Mobis, 1625344@mobis.co.kr, 일반회원

° Corresponding Author : (ORCID:0000-0002-7630-8533)School of Electrical Engineering, Korea Advanced Institute of Science and Technology, seungchan.lim@kaist.ac.kr, 학생회원

\* School of Electrical Engineering, Korea Advanced Institute of Science and Technology, hcpark@kaist.ac.kr, 중신회원

논문번호 : KICS2017-11-348, Received November 14, 2017; Revised January 22, 2018; Accepted March 14, 2018

으로써 활용하고자 하는 무선 캐싱(wireless caching) 기술이 하나의 대안으로서 주목받고 있다<sup>2-5)</sup>.

무선 캐싱 시스템에서는 대용량 멀티미디어 파일들을 수요가 적은 시간에 미리 저장한다. 인기 있는 데이터의 사전 저장을 통하여 트래픽 수요가 많아지는 시간대의 백용 사용 부담을 줄이고 사용자들의 반복 요청에 대해 효과적으로 대응할 뿐만 아니라 서비스 전송 지연시간을 감소시킬 수 있다는 장점을 가진다<sup>2)</sup>. 따라서 무선 캐싱 시스템이 가지는 장점을 극대화하기 위해서는 제한된 캐싱 용량에 대하여 데이터를 효율적으로 저장하고 분배하는 방법에 대한 연구가 중요하다<sup>4)</sup>.

무선 캐싱 시스템은 파일 분배 방법에 따라 파일 전체를 저장하는 단일 파일 캐싱 시스템(single file caching systems)과 파일을 여러 장치에 분산하여 저장하는 분산 캐싱 시스템(distributed caching systems)로 구분될 수 있다<sup>5-6)</sup>. 이 중 분산 캐싱 시스템은 단일 파일 캐싱 시스템에 비해 사용자가 파일을 성공적으로 수신할 확률이 높고 제한된 저장 장치를 효율적으로 사용할 수 있다는 장점 때문에 최근 큰 주목을 받고 있다<sup>7-10)</sup>. 기존 분산 캐싱 시스템에서는 MDS (maximum distance separable) 부호화된 파일의 분산 저장 및 송수신을 가정하였다. 이러한 시스템에서는 MDS 부호화된 파일을 여러 소형 기지국에 분산하여 저장하여도 메시지 길이만큼의 부호어를 수신하는 경우 항상 성공적인 복호가 가능하기 때문에 효율적인 시스템 운용이 가능하다. 하지만, 일반적으로 MDS 부호는 부/복호화 복잡도가 높고 임의의 부호 파라미터를 유연히 지원하지 못한다는 단점을 가지고 있다.

본 논문에서는 분산 부호화 캐싱 시스템에 대하여 상대적으로 더욱 실용적인 채널 부호의 적용을 고려한다. 기존 연구에서 고려하였던 MDS 부호는 수신된 부호어의 길이에 따라서 복호 실패 확률이 0 또는 1로써 결정되지만, 일반적인 채널 부호를 적용한다면 저장된 부호어 길이에 의존하는 복호 실패 확률이 존재하게 된다<sup>13-14)</sup>. 복호 실패 사건은 결국 시스템의 서비스 실패를 야기하므로 파운틴 류의 채널 부호를 고려하는 경우 시스템에 대한 최적의 파일 저장 기법이 달라질 수 있다. 본 논문에서는 DVB-H 표준 등에서 사용되는 램터 부호<sup>15-16)</sup> 등의 파운틴(fountain) 부호의 적용을 고려하고 분산 부호화 캐싱 시스템의 서비스 실패 확률을 정의하여 이를 분석한다. 또한 분석된 결과에 기반하여 시스템의 서비스 실패 확률을 최소화할 수 있는 파일 분산 저장 기법을 제안함으로써 높은

신뢰도를 갖는 시스템의 설계를 목표한다.

본 논문의 구성은 다음과 같다. 2장에서는 분산 부호화 저장 시스템 모델 및 시스템의 동작을 설명한다. 3장에서는 적용된 채널 부호의 복호 실패 확률을 반영하여 서비스 실패 확률을 정의하고 시스템에 대한 최적의 파일 저장 기법을 도출하는 문제를 정의한다. 4장에서는 정의된 서비스 실패 확률의 하한 값을 분석하고 이를 달성하기 위한 파일 분산 저장 알고리즘을 제안한다. 5장에서는 모의 실험을 통하여 제안 기법의 우수성을 검증하고 6장에서 본 논문을 요약 및 정리한다.

## II. 분산 부호화 캐싱 시스템

### 2.1 시스템 모델

그림 1은 하나의 서버(server)와 다수의 소형 기지국(helper) 및 사용자로 구성된 분산 부호화 캐싱 시스템을 보여준다. 서버는 총  $F$  개의 파일로 구성된 라이브러리를 보유하고 있으며 모든 파일의 크기는  $K$  로써 동일함을 가정한다. 사용자가 라이브러리 중 특정 파일을 소형 기지국에 요청하면  $H$  개의 소형 기지국으로부터 요청 파일에 대한 부호어를 수신한다. 이때, 분할된 부호어를 사용자에게 전송하는 소형 기지국들을 복호 집합(decoding set)이라고 하며 각 소형 기지국의 캐시 용량은  $M$ 의 크기를 가진다.

라이브러리에 저장되어 있는  $1 \leq i \leq F$  번째 파일은  $K$ 의 크기를 갖고,  $\kappa$ 개의 프레임으로 구성된다. 서버는  $k = K/\kappa$ 로써 동일한 크기를 갖는 각 프레임에 대하여 부호화를 수행 후 부호화된 파일들을 소형 기지국에 분산 저장한다.

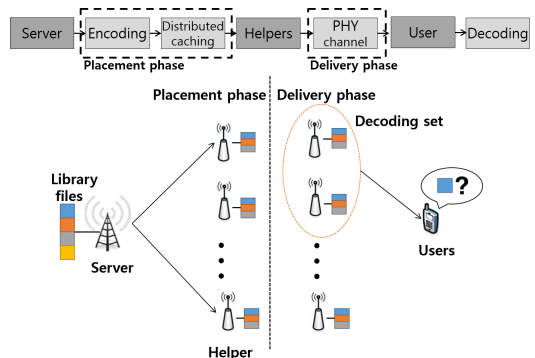


그림 1. 분산 부호화 캐싱 시스템의 저장 단계와 전송 단계 동작  
Fig. 1. Placement and delivery phases of distributed coded caching systems

각  $i$  번째 파일에 대하여 부호화된 프레임의 크기를  $n_i$ 로 나타낼 때, 모든 파일에 대한 프레임 크기를 길이가  $F$ 인 튜플(tuple)  $\mathbf{n} = [n_1, \dots, n_F]$ 로써 표현할 수 있다. 사용자는  $1 \leq i \leq F$  번째 파일을 확률  $q_i$ 로 요청하게 되며 모든 파일의 확률을 튜플  $\mathbf{q} = [q_1, \dots, q_F]$ 로써 표기한다. 모든 파일의 인덱스는 요청확률에 따라 내림차순으로 정렬되어 있음을 가정하며 ( $q_i \geq q_j, i \leq j$ ) 모든 파일에 대한 요청 확률의 합은 1이다.

### 2.2 시스템의 동작

분산 부호화 캐싱 시스템은 저장 단계(placement phase)와 전송 단계(delivery phase)로 구별되어 동작한다. 시스템의 저장 단계는 서버가 소형 기지국들의 캐시에 부호어들을 분산 저장하는 사전 단계를 의미한다. 이 단계에서 서버는 총  $F$  개의 라이브러리 파일 중 실제 저장될  $f$  개의 파일을 결정하고, 각 파일에 대하여 부호화 프레임들을 소형 기지국에 분산 저장한다. 이 때, 한 파일을 구성하는 각 부호어는 중복되어 저장되지 않음을 가정한다. 전송 단계에서는 사용자가 요청한 파일을 소형 기지국들로부터 수신 받는다. 사용자는 복호 집합에 포함되는 소형 기지국들로부터 부호어를 수신 받아 프레임에 대한 복호 후 해당 파일을 재구성한다.

서버의 라이브러리에 저장된  $1 \leq i \leq F$  번째 파일은  $(n_i, k)$  부호를 통하여  $n_i$ 의 길이를 가지는  $\kappa$ 개의 부호어로 부호화된 후  $H$  개의 소형 기지국에 분산 저장된다. 기존 연구에서는 MDS 부호를 고려하여 부호어 길이가 파일의 크기 이상인 경우 ( $n_i \geq k$ ) 항상 복호에 성공한다는 가정을 반영하였다. 하지만 MDS 부호는 부호 파라미터를 유동적으로 설정하기 어렵고 부/복호화 복잡도가 높기 때문에 실제 시스템에 적용이 힘들다는 단점을 가진다. 본 논문에서는 랩터 부호 등의 파운틴 부호의 적용을 고려함으로써 보다 실용적인 부호화 캐싱 시스템에 대한 성능 분석 및 최적의

표 1. 다양한  $(n_i, k)$  부호에 대한 복호 실패 확률  $P(n_i)$   
Table 1. Decoding failure probability  $P(n_i)$  for  $(n_i, k)$  codes

	Condition	
	$n_i \geq k$	$n_i < k$
MDS codes	0	1
Random fountain codes <sup>[13]</sup>	$0.5^{n_i-k}$	1
Raptor codes (RFC-5053) <sup>[16]</sup>	$0.85 \times 0.567^{n_i-k}$	1

파일 분산 배치 알고리즘을 연구한다. 시스템에 적용 가능한 채널 부호를 고려할 때,  $i$  번째 파일에 대한 복호 실패 확률  $P(n_i)$ 을 표 1과 같이 나타낼 수 있다.

## III. 최적의 파일 저장 문제

### 3.1 서비스 실패 확률

본 논문에서는 분산 부호화 캐싱 시스템의 성능 지표로써 서비스 실패(service failure) 확률을 고려한다. 서비스 실패 사건은 사용자로부터 요청된 파일이 소형 기지국에 저장되지 않았기 때문에 발생하는 간과(missing) 사건과 사용자가 소형 기지국으로부터 수신 받은 프레임으로부터 복호에 실패하는 경우 발생하는 복호 실패(decoding failure) 사건으로 구분된다.

특정  $1 \leq i \leq F$  번째 파일에 대한 간과 사건은 소형 기지국에 저장되어 있는 파일의 길이가 0인 경우에만 발생하므로 간과 확률  $P_{\text{miss}}(n_i)$ 을 다음과 같이 표현할 수 있다.

$$P_{\text{miss}}(n_i) = \mathbb{I}(n_i = 0) \cdot 1. \quad (1)$$

여기서  $\mathbb{I}(a)$ 은 표지함수(indicator function)로써  $a$ 의 참 또는 거짓 여부에 따라 각각 1 또는 0을 출력한다.

복호 실패 확률은 시스템에 적용된 부호화 기법에 따라서 표 1과 같이 달라질 수 있다. 만약 부호어의 길이가 프레임의 길이보다 작다면 사용자는 항상 수신 파일의 복호에 실패하게 되므로 캐싱 시스템의 합리적인 운영 관점에서 이러한 사건은 배제함을 가정한다. 반대로 부호어 길이가 프레임의 길이 이상인 경우 복호 실패 확률은 부호어 길이  $n_i$ 의 함수로써 표현되므로  $i$  번째 파일에 대한 복호 실패 확률  $P_{\text{dec}}(n_i)$ 을 다음과 같이 표현한다.

$$P_{\text{dec}}(n_i) = \mathbb{I}(n_i \geq k) \cdot P(n_i). \quad (2)$$

식 (1)과 (2)에서 고려한 간과 사건과 복호 실패 사건은 상호 배타적으로 발생하므로 서비스 실패 확률은 간과 확률 및 복호 실패 확률의 합으로써 표현된다. 따라서 분산 부호화 캐싱 시스템의 모든 파일에 대한 평균 서비스 실패 확률  $\bar{P}(\mathbf{n})$ 은 각 파일의 요청 확률을 반영하여 다음과 같이 정의된다.

$$\bar{P}(\mathbf{n}) = \sum_{i=1}^F q_i [\mathbb{I}(n_i \geq k) \cdot P(n_i) + \mathbb{I}(n_i = 0) \cdot 1]. \quad (3)$$

### 3.2 평균 서비스 실패 확률을 최소화하는 파일 분산 저장 문제

식 (3)으로 표현된 평균 서비스 실패 확률은 라이브러리 파일에 대한 프레임 크기 튜플  $\mathbf{n}$ 에 대한 함수이므로 시스템의 성능이 파일의 분산 저장 방법에 의존함을 알 수 있다. 본 논문에서는 신뢰도 높은 분산 부호화 캐싱 시스템의 설계를 위해서 평균 서비스 실패 확률을 최소화하는 파일 분산 저장 문제를 다음과 같이 정의한다:

$$\begin{aligned} \mathbf{n}^* &= \operatorname{argmin}_{\mathbf{n}} \bar{P}(\mathbf{n}) \\ \text{subject to } & \sum_{i=1}^F \kappa n_i \leq HM \\ & n_i \in \mathbb{Z}^+, 1 \leq i \leq F \end{aligned} \quad (4)$$

식 (4)의 최적화 문제에서 시스템이 저장 가능한 총 용량은  $HM$ 으로 제한되며, 저장되는 부호어의 길이는 음이 아닌 정수로서 표현된다. 식 (4)의 문제는 정수 문제(integer problem)로서 최적의 해를 찾기가 매우 어려우므로 최적의 파일 분산 저장 문제를 해결하기 위해서는 다른 방식의 접근이 필요하다.

### IV. 평균 서비스 실패 확률을 최소화하는 파일 분산 저장 알고리즘

본 장에서는 식 (4)에 표현된 최적의 파일 분산 저장 기법의 성능에 근접하는 파일 분산 저장 알고리즘을 제안한다. 우수한 평균 서비스 실패 확률을 달성 가능한 알고리즘을 설계하기 위하여 평균 서비스 실패 확률에 대한 하한 값을 분석하고 분석된 하한 값을 달성할 수 있는 파일 분산 저장 방법을 모색한다.

#### 4.1 평균 서비스 실패 확률에 대한 하한 값 분석

라이브러리에 보유된 총  $F$  개의 파일 중 실제 저장할  $f \leq F$  개의 파일에 대한 인덱스 집합을  $S$ 라고 할 때, 식 (3)의 평균 서비스 실패 확률을 다음과 같이 재 표현할 수 있다:

$$\begin{aligned} \bar{P}(\mathbf{n}) &= \sum_{i \in S} q_i P(n_i) + \sum_{i \in S^c} q_i \cdot 1 \\ &\geq \sum_{i \in S} q_i \end{aligned}$$

상기 식은 평균 서비스 실패 확률이 저장되지 않는 파일들의 요청 확률의 합보다 작아질 수 없음을 의미한

다. 따라서 라이브러리 중 일부  $f \leq F$  개의 파일을 저장하는 경우, 요청 확률이 높은 파일부터 순차적으로 저장하고 최소의 요청 확률을 갖는  $(F-f)$  개의 파일을 저장하지 않는 것이 서비스 실패 확률을 낮출 수 있음을 알 수 있다. 이러한 사실로부터 다음과 같이 평균 서비스 실패 확률에 대한 하한 값을 분석할 수 있다:

$$\begin{aligned} \bar{P}(\mathbf{n}) &= \sum_{i=1}^F q_i [\mathbb{I}(n_i \geq k) \cdot P(n_i) + \mathbb{I}(n_i = 0) \cdot 1] \\ &\geq f \left( \prod_{i=1}^f q_i P(n_i) \right)^{1/f} + \sum_{i=f+1}^F q_i \\ &\geq \sum_{i=1}^f q_i P(n_i) + \sum_{i=f+1}^F q_i \end{aligned} \quad (5)$$

식 (5)는 각 파일에 대한 요청 확률 및 복호 실패 확률의 곱에 대한 산술-기하 평균 부등식을 적용함으로써 유도되었다. 식 (5)에 표현된 하한 값을  $\overline{P}_{LB}(\mathbf{n})$ 로써 정의할 때, 이를 다음과 같이 재 표현할 수 있다:

$$\begin{aligned} \overline{P}_{LB}(\mathbf{n}) &= f \left( \prod_{i=1}^f q_i P(n_i) \right)^{1/f} + \sum_{i=f+1}^F q_i \\ &= f \lambda(f) + \sum_{i=f+1}^F q_i \end{aligned} \quad (6)$$

여기서  $\lambda(f)$ 는 기하 평균 값을 나타낸다.

**예시:** 분산 부호화 캐싱 시스템에 적용된 채널 부호를 DVB-H 표준에서 사용되고 있는 램터 부호(RFC-5053)라고 가정하자. 표 1에 나타난 램터 부호의 복호 실패 확률을 반영하여 식 (6)의 하한 값을 다음과 같이 계산할 수 있다:

$$\overline{P}_{LB}(\mathbf{n}) = f \underbrace{\left( \prod_{i=1}^f q_i \right) 0.85^f \times 0.567^{\frac{HM}{\kappa} - fk}}_{\lambda(f)}^{1/f} + \sum_{i=f+1}^F q_i$$

상기 식에서 부호어 길이의 합은 저장 장치의 크기에 의해서 제한되므로 저장 장치를 최대한으로 활용함으로써 기하 평균에 해당되는 항인  $\lambda(f)$ 를 계산할 수 있다.

### 4.2 제안 파일 분산 저장 알고리즘

평균 서비스 실패 확률에 대한 하한 값  $\overline{P_{LB}}(\mathbf{n})$ 을 달성하기 위해서는 모든 파일  $1 \leq i \leq f$ 에 대하여 요청 확률  $q_i$  및 복호 실패 확률  $P(n_i)$ 의 곱이 기하 평균  $\lambda(f)$ 과 동일하도록 부호어 길이가 결정되어야 한다. 주어진 파일 저장 개수  $f$ 에 대하여 파일의 요청 확률 분포  $q_f$ 가 주어진다면 기하 평균  $\lambda(f)$ 를 구할 수 있다. 따라서 저장할  $i$  번째 파일에 대한 부호어 길이를 계산된 기하 평균  $\lambda(f)$ 로부터 다음과 같이 설정한다:

$$n_i = \left\lfloor P^{-1}\left(\frac{\lambda(f)}{q_i}\right) \right\rfloor.$$

여기서  $P^{-1}(\cdot)$ 는 채널 부호의 복호 실패 확률  $P(\cdot)$ 에 대한 역함수를 나타낸다. 이와 같이 결정된 부호어 길이에 대하여 총 캐싱 용량에 대비 남은 용량에 대한 추가적인 할당 과정을 수행한다. 총 캐싱 용량을 넘지 않는 범위에서 파일 요청 확률이 큰 파일부터 추가적으로 한 비트 씩 부호어 길이를 증가시킴으로써 남은 캐싱 용량을 모든 파일에 대하여 재할당한다.

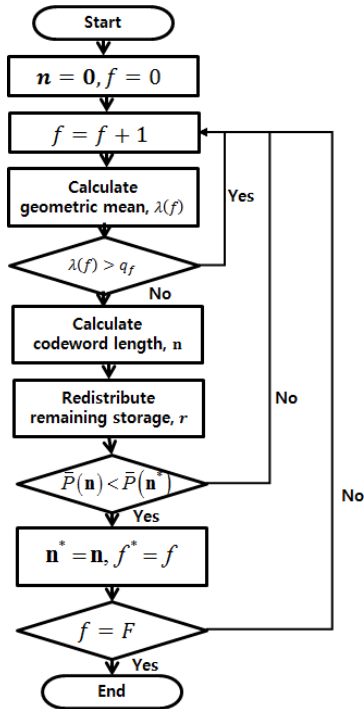


그림 2. 제안 알고리즘  
Fig. 2. Proposed algorithm

$$n_i = \left\lfloor P^{-1}\left(\frac{\lambda(f)}{q_i}\right) \right\rfloor + \alpha_i \quad (\alpha_i \in 0,1).$$

상기 과정은 저장하는 파일의 개수  $f$ 가 결정되었을 때를 가정하였으므로 캐싱에 저장할 파일의 수  $f$ 를 추가적으로 최적화해야 한다. 모든  $1 \leq i \leq f$ 에 대하여 위의 과정을 반복하며 최소의 평균 서비스 실패 확률을 갖는 저장 파일 수를 결정할 수도 있으나, 판별식에 대한 조사를 수행함으로써 불필요한 연산을 생략할 수 있다. 임의의  $1 \leq f \leq F$ 에 대하여 요청 확률  $q_f$ 가 기하평균  $\lambda(f)$ 보다 작은 경우에는  $q_f P(n_f)$ 와 기하평균의 차이가 증가하므로 해당 파일을 저장하지 않는 것이 평균 서비스 실패 확률을 낮추는 데 도움이 된다. 따라서 다음의 부등식을 고려하고 임의의  $f$ 에 대하여 부등식이 성립하는 경우에는 파일 수  $f$ 를 최적화 대상에서 제외시킬 수 있다.

$$\lambda(f) > q_f \tag{7}$$

식 (7)의 판별식은 임의의 저장 파일의 개수  $f$ 에 대하여 추가적인 계산 없이 파일의 저장 여부를 결정할 수 있게 해준다. 고려해야하는 모든 파일 수에 대한 범위를  $f_{range}$ 라 표기할 때,  $f_{range}$ 에 속하는 모든 파일 수  $f$ 에 대하여 파일에 대한 분산 저장 분포  $\mathbf{n}$ 을 구하고 이에 대한 평균 서비스 실패 확률을 최소화하도록 저장 파일의 부호어 길이를 결정한다. 제안 파일 분산 저장 알고리즘은 그림 2로써 요약되어 있다.

### V. 모의 실험

본 장에서는 제안된 파일 분산 저장 기법의 성능을 모의 실험을 통하여 검증한다. 모의 실험을 위하여 분산 부호화 캐싱 시스템에 대해 설정된 파라미터들은 표 2에 나타나 있다. 사용자와 소형기지국 간에 이상적인 채널 환경을 고려하여 데이터 송수신에 오류가 없음을 가정하였고, 비디오 파일 요청 확률 분포로써

표 2. 모의 실험 환경  
Table 2. Simulation environment

Parameters	Specification
Zipf exponent, $\gamma$	0 ~ 3
Total overhead, $O$	0 ~ 20 %
File size, $K = \kappa k$	30 MB
Frame size, $k$	1024 bits
Number of files, $F$	20

지프(Zipf) 분포를 고려하여 파일에 대한 요청 확률을 다음과 같이 설정하였다<sup>6)</sup>:

$$q_i = \frac{1/i^\gamma}{\sum_{j=1}^F 1/j^\gamma}$$

여기서 지프 분포의 지수(Zipf exponent)  $\gamma$ 는 특정 파일에 대한 인기도의 치우침을 나타내는 파라미터로써 그림 3에서 나타난 바와 같이  $\gamma$  값이 커질수록 일부 파일에 요청이 집중되고 작아질수록 모든 파일에 대한 요청 확률이 균일해진다. 분산 부호화 캐싱 시스템에 대하여 랩터 코드(RFC-5053)를 적용하였으며<sup>16)</sup>, 전체 오버헤드  $O$ 를 다음의 식으로 정의하여 시스템 파라미터로써 고려하였다:

$$O = \frac{\sum_{i=1}^F n_i - FK}{\sum_{i=1}^F n_i} \times 100 \quad [\%]$$

기존 문헌에서 고려되었던 MDS 부호가 적용된 분산 캐싱 시스템<sup>7-10)</sup>의 경우 표 2의 모의 실험 환경을 반영한 성능 평가가 어려운 실정이다. 따라서 제안된 파일 분산 저장 기법(Proposed)에 대한 비교 대상으로서 다음의 일반적인 저장 기법을 고려하였다.

- **균일 저장 기법(Uniform):** 모든 파일에 대하여 같은 길이의 부호어를 할당하는 방법
- **비율적 저장 기법(Proportional):** 파일 오버헤드를 사용자의 파일 요청 확률에 비례하여 결정하는 방법

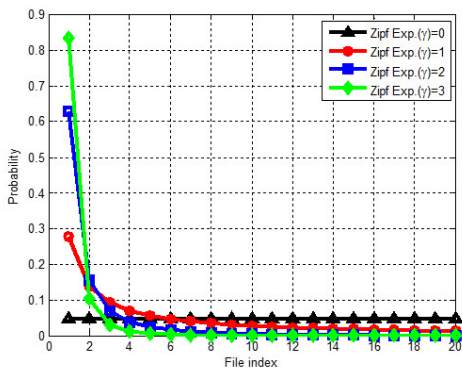
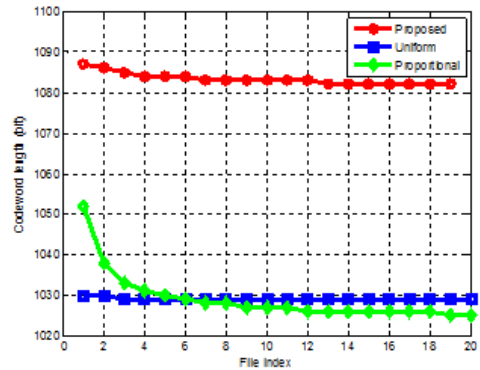


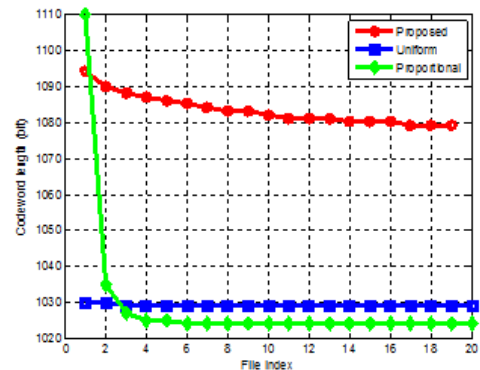
그림 3. 지프 분포  
Fig. 3. Zipf distribution

그림 4는 제안 알고리즘에 따라 결정된 부호어 길이에 대한 분포를 보여준다. 제안 기법은 경우에 따라서 가장 인기도가 낮은 파일을 저장하지 않으므로써 사용 가능한 캐싱 용량을 다른 파일의 오버헤드로 할당하는 것을 알 수 있으며 이러한 방법이 시스템의 평균 서비스 실패 확률 최소화 관점에서 유리하다는 것을 알 수 있다. 또한, 제안 알고리즘을 수행 시 파일의 인기도가 높아짐에 따라 상대적으로 저장되는 부호어의 길이가 길어진다는 사실을 관찰 할 수 있다.

그림 5는 시스템의 전체 오버헤드 증가에 따른 시스템의 평균 서비스 실패 확률을 보여준다. 저장 용량이 늘어남에 따라 저장되는 부호어의 길이가 길어지게 되므로 서비스 실패 확률이 낮아지는 것을 확인할 수 있으며, 다른 저장 기법 대비 제안된 분산 파일 저장 기법이 우수한 성능을 갖는다는 것을 확인할 수 있다. 결과 그래프에서 서비스 실패 확률이 수평적인 형태로 나타나는 구간을 확인할 수 있는데 이는 20번째 파일의 인기도와 유사한 값을 갖는 것을 확인할 수



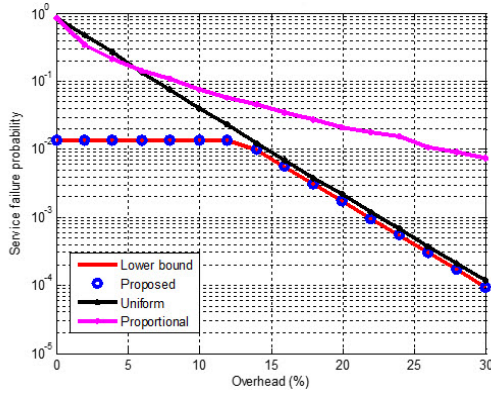
(a)  $\gamma = 1$



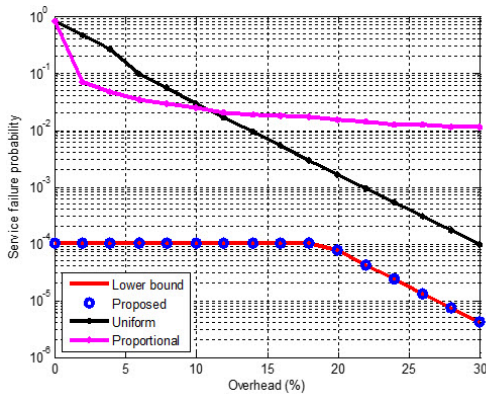
(b)  $\gamma = 3$

그림 4. 제안 알고리즘의 부호어 길이 분포  
Fig. 4. Distribution of the codeword lengths for the proposed algorithm





(a)  $\gamma = 1$



(b)  $\gamma = 3$

그림 5. 시스템 오버헤드에 따른 평균 서비스 실패 확률  
Fig. 5. Average service failure probability for total overhead

있었고, 이 구간은 인기도가 제일 낮은 파일을 저장하지 않고 나머지 파일의 신뢰도를 높이는 것이 시스템의 성능 향상에 유리한 구간을 의미한다. 또한 이 구간 이후에는 오버헤드가 증가함에 따라서 서비스 실패 확률이 감소하는 기술품이 균일 저장 기법과 동일함을 확인할 수 있는데, 이러한 현상은 각 파일의 길이가 일정 수준 이상이 된다면 모든 파일에 대한 부호어 길이가 균일하게 증가되고 있음을 의미한다.

그림 6은 지프 분포의 지수가 증가함에 따라 시스템의 평균 서비스 실패 확률의 변화를 보여준다. 제안 기법을 이용하여 시스템을 설계 하는 경우 다른 기법 대비 매우 낮은 확률을 달성 가능할 뿐만 아니라 식 (5)에 분석된 하한 값에 매우 근접한 성능을 달성하는 것을 관찰할 수 있다. 지프 분포 지수가 0에 근접한 경우에는 균일 저장 기법이 서비스 실패 확률을 최소화하는 방법임을 확인할 수 있지만, 지프 분포 지수가 증가함에 따라 파일의 인기도를 적절히 반영하지 못

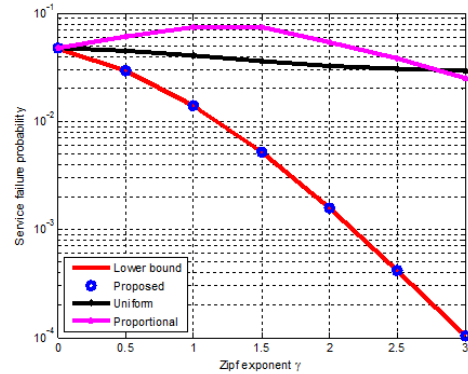


그림 6. 지프 분포 지수에 따른 평균 서비스 실패 확률 ( $O=10\%$ )  
Fig. 6. Average service failure probability for Zipf exponent ( $O=10\%$ )

하는 균일 저장 기법은 하한 값에서 멀어진다는 것을 알 수 있다. 반면 비율적 저장 기법은 높은 인기도를 가지는 파일의 복호 실패 확률을 낮추기 위해 상대적으로 낮은 인기도 파일의 복호 실패 확률이 높아지기 때문에 전체적인 서비스 실패 확률은 증가하는 현상이 발생한다. 제안하는 저장 기법은 지프 분포의 지수가 증가함에 따라서도 하한 값을 달성함을 알 수 있으며 지프 분포의 지수가 증가함에 따라 비교 대상 기법 대비 제안 기법의 성능 이득이 더욱 커지는 것을 확인할 수 있다.

## VI. 결론

분산 부호화 캐싱 시스템은 저장 장치를 활용한 파일 저장 방법에 따라서 시스템의 성능이 결정된다. 기존 연구에서는 MDS 부호화 시스템에 대한 파일 저장 기법을 연구하였지만, 보다 실용적인 파운틴 류의 부호를 고려하는 경우 최적의 파일 저장 기법이 달라질 수 있다. 본 논문에서는 분산 부호화 캐싱 시스템의 평균 서비스 실패 확률을 최소화하는 파일 저장 기법을 제안하였다. 채널 부호가 가지는 복호 실패 확률을 반영하여 시스템의 서비스 실패 확률을 정의하고 이에 대한 하한 값을 분석하였다. 또한, 시스템의 서비스 실패 확률을 최소화하는 파일 저장 알고리즘을 제안하고, 제안 기법의 성능을 모의실험을 통하여 분석하였다. 제안 기법은 다른 기법 대비 매우 낮은 서비스 실패 확률을 가질 뿐 아니라 분석된 하한 값에 매우 근접한 성능 가지는 사실을 모의실험을 통해서 검증하였다.

References

- [1] Cisco, *Cisco Visual Networking Index Global Mobile Data Traffic Forecast update*, 2015-2020.
- [2] J. G. Andrews, "Seven ways that HetNets are a cellular paradigm shift," *IEEE Commun. Mag.*, vol. 51, no. 3, pp. 136-144, Mar. 2013.
- [3] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131-139, Feb. 2014.
- [4] U. Niesen, D. Shah, and G. W. Wornell, "Caching in wireless networks," *IEEE Trans. Inf. Theory*, vol. 58, no. 10, pp. 6524-6540, May 2012.
- [5] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402-8413, May 2013.
- [6] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142-149, Apr. 2013.
- [7] Y. Ugur, Z. H. Awan, and A. Sezgin, "Cloud radio access networks with coded caching," *WSA 2016; 20th Int. ITG Workshop on Smart Antennas*, pp. 1-5, Mar. 2016.
- [8] V. Bioglio, F. Gabry, and I. Land, "Optimizing MDS codes for caching at the edge," in *Proc. 2015 IEEE GLOBECOM*, pp. 1-6, Dec. 2015.
- [9] B. Hong and W. Choi, "Optimal storage allocation for wireless cloud caching systems with a limited sum storage capacity," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6010-6021, Sept. 2016.
- [10] B. Hong and W. Choi, "Study on wireless distributed storage based on MDS coding," in *Proc. Symp. Korean Inst. Commun. and Inf. Sci.*, 2016. 6, pp. 507-508.
- [11] A. Liu, V. Lau, F. Zhuang, and J. Chen, "Mixed timescale cross-layer optimization for multi-antenna D2D networks," in *Proc. 2015 IEEE GLOBECOM*, pp. 1-6, Dec. 2015.
- [12] A. Liu and V. K. N. Lau, "Cache-enabled opportunistic cooperative MIMO for video streaming in wireless systems," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 390-402, Jan. 2014.
- [13] D. J. C. MacKay, "Fountain codes," *IEE Proc. Commun.*, vol. 152, no. 6, pp. 1062-1068, Dec. 2005.
- [14] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Sci.*, 2002, pp. 271-280, 2002.
- [15] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2551-2567, Jun. 2006.
- [16] T. Stockhammer, A. Shokrollahi, M. Watson, M. Luby, and T. Gasiba, *Application layer forward error correction for mobile multimedia broadcasting*, Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T and Media FLO, CRC Press, pp. 239-280, May 2008.

이 원 석 (Wonseok Lee)



2015년 2월 : 성균관대학교 전자전기공학부 학사  
 2017년 8월 : 한국과학기술원 전기 및 전자공학부 석사  
 2017년 1월~현재 : 현대모비스 <관심분야> 무선 통신, 통신공학, 채널 부호

임 승 찬 (Seung-Chan Lim)



2011년 2월 : 홍익대학교 전자전기공학과 학사  
 2013년 2월 : 포항공과대학교 전자전기공학과 석사  
 2015년 3월~현재 : 한국과학기술원 전기 및 전자공학부 박사과정

<관심분야> 다중 접속, 다중 안테나, 채널부호



**박 현 철 (Hyuncheol Park)**



1983년 2월 : 연세대학교 전자  
공학과 학사

1985년 2월 : 연세대학교 전자  
공학과 석사

1997년 8월 : Georgia Institute  
of Technology Electrical  
Eng. 박사

1984년 12월~2002년 1월 : 삼성전자 수석연구원

2002년 2월~2009년 2월 : 한국정보통신대학교(ICU)  
부교수

2014년 11월~2016년 11월 : 미래창조과학부 이동통  
신 CP

2009년 3월~현재 : 한국과학기술원 정교수

<관심분야> 이동통신 시스템