

# 소형셀 환경에서 사용자 컨텍스트 기반 무선 캐시 알고리즘

정현기\*, 정소이\*, 이동학\*, 이승규\*\*, 김재현<sup>o</sup>

## Wireless Caching Algorithm Based on User's Context in Smallcell Environments

Hyun Ki Jung\*, Soyi Jung\*, Dong Hak Lee\*, Seung Que Lee\*\*, Jae-Hyun Kim<sup>o</sup>

### 요약

본 논문에서는 home 소형셀 대비 넓은 커버리지를 갖고 많은 사용자를 서비스 하는 enterprise/urban 소형셀 환경에서 적용할 수 있는 사용자 컨텍스트 기반 캐시 알고리즘을 제안한다. 소형셀 캐시 기법은 소형셀 사용자의 웹 트래픽을 소형셀 내부에 위치한 저장 공간에 저장하는 방법으로 코어망 트래픽을 감소시키는 효과가 있다. 본 논문에서는 기존의 알고리즘과 달리 Mobile Edge Computing(MEC)의 개념을 적용하여 소형셀 내부가 아닌 edge server에 사용자 트래픽을 캐시하며 사용자 특성을 반영하기 위해 사용자를 그룹화한다. 또한, 그룹별 저장 공간의 크기를 달리하고, 캐시 업데이트 주기를 캐시 적중률에 따라 변경하여 코어망으로부터 제공받는 트래픽을 감소하고자 하였다. 성능 분석 결과 기존 알고리즘 대비 캐시 적중률 측면에서 약 11%, cache efficiency 측면에서 약 5.5%의 성능 향상을 확인할 수 있었다.

**Key Words** : Smallcell Cache, Local Cache, User's Context, Offloading, Mobile Edge Computing

### ABSTRACT

In this paper, we propose a cache algorithm based on user's context for enterprise/urban smallcell environments. The smallcell caching method is to store mobile users' data traffic at a storage which is equipped in smallcell base station and it has an effect of reducing core networks traffic volume. In our algorithm, contrary to existing smallcell cache algorithms, the cache storage is equipped in a edge server by using a concept of the Mobile Edge Computing. In order to reflect user's characteristics, the edge server classifies users into several groups based on user's context. Also the edge server changes the storage size and the cache replacement frequency of each group to improve the cache efficiency. As the result of performance evaluation, the proposed algorithm can improve the cache hit ratio by about 11% and cache efficiency by about 5.5% compared to the existing cache algorithm.

※ 본 연구는 미래창조과학부의 정부출연금사업의 일환으로 수행하였음. [15Z11110, 트래픽 용량 증대를 위한 액세스 네트워크 원천기술 연구]

• First Author : Ajou University Department of Electrical and Computer Engineering, happy8076@ajou.ac.kr, 학생회원

<sup>o</sup> Corresponding Author : Ajou University Department of Electrical and Computer Engineering, jkim822@ajou.ac.kr, 종신회원

\* Ajou University Department of Electrical and Computer Engineering, sogloomy@ajou.ac.kr, inspire109@ajou.ac.kr, 학생회원

\*\* Electronics and Telecommunications Research Institute, sqlee@etri.re.kr, 정회원

논문번호 : KICS2016-04-077, Received April 30, 2016; Revised June 8, 2016; Accepted June 8, 2016

## I. 서 론

2020년 상용화를 목표로 하는 5G 이동통신에서는 기존 4G 이동통신 대비 향상된 성능을 요구하고 있으며 서비스 측면에서도 8K-UHD, 홀로그램, 3D 영상과 같은 초고용량 영상 콘텐츠 제공, 초 실시간 서비스, IoT(Internet of Things) 서비스 제공을 목표로 하고 있다<sup>1,2)</sup>. 이에 따라 4G 이동통신 대비 5G 이동통신에서의 모바일 트래픽은 급증할 것으로 예측되며 Cisco는 2015년 대비 2020의 모바일 트래픽이 약 8배 증가한 30.6 EByte가 될 것으로 예측하였다<sup>3)</sup>.

급증하는 모바일 트래픽 수용을 위해 Heterogeneous 네트워크(HetNet) 기술, advanced MIMO/beamforming 기술, CoMP(Coordinated Multi-point)들이 연구 중에 있으며 향후 통신사업자들은 과도한 투자/운영 비용을 줄이기 위해 HetNet 기술을 적극 활용할 것으로 예상된다<sup>4,5)</sup>. 이러한 HetNet 기술의 핵심 구성 요소인 소형셀은 기존 매크로 기지국에 비해 낮은 전송 파워와 좁은 커버리지를 갖는 기지국으로 셀 용량 증대를 통해 급증하는 트래픽 수용에 도움이 될 것으로 예상된다.

하지만 소형셀 배치를 통한 트래픽 수용은 코어망의 부하를 증가시키며, 낮은 전송 용량을 갖는 소형셀의 백홀 링크 또한 문제가 된다<sup>6)</sup>. 따라서 코어망 부하 감소 및 소형셀 백홀 링크 문제 해결을 위해 소형셀 환경에서 코어망으로 집중되는 트래픽을 감소시킬 필요가 있으며 이를 해결하기 위한 방법으로 최근 소형셀 캐시 기술이 연구 중에 있다.

소형셀 캐시 기술이란 코어망 부하 감소를 위한 방법 중 하나로 소형셀 내부에 저장 공간을 위치시켜 소형셀 사용자의 웹 데이터 트래픽을 캐시하는 기술이다. 이를 통해 소형셀 내부에 위치한 저장 공간에 캐시된 콘텐츠는 코어망을 통하지 않고 사용자에게 직접 전송될 수 있다. 기존 연구<sup>6-8)</sup>에서는 소형셀 사용자의 특성을 반영하지 않고 일반적인 사용자의 인기 콘텐츠를 효율적으로 배치하는 기법에 대한 연구가 수행되었다. 또한, 사용자 특성을 반영하고자 연구한 논문<sup>11)</sup>에서는 집, 작은 사무실에 설치되는 home 소형셀 환경만을 가정하고 있어서 넓은 범위에 설치되어 다양한 사용자를 서비스하는 enterprise/urban 소형셀 환경에서는 적용하기 어렵다는 단점이 있다.

따라서 본 논문에서는 enterprise/urban 소형셀 환경에서 사용자 grouping을 통해 사용자 특성을 반영하고 Mobile Edge Computing(MEC)의 개념을 활용하여 소형셀이 아닌 edge server에 캐시 하는 소형셀

캐시 알고리즘을 제안하고자 한다.

## II. 관련 연구

### 2.1 캐시 교체 정책

캐시 저장 공간의 크기는 제한적이므로 저장 공간에 존재하는 콘텐츠는 특정 기준에 따라 교체를 해야만 캐시 적중률을 높일 수 있다. 이에 따라 오래전부터 캐시 교체 정책에 관한 연구는 진행되었으며 다음과 같은 캐시 교체 정책들이 있다.

먼저, LRU(Least Recently Used) 방식은 캐시된 콘텐츠 중 최근 가장 오랫동안 사용하지 않은 콘텐츠를 먼저 캐시 저장 공간에서 삭제하고 새로운 콘텐츠를 저장하는 방식으로 일반적으로 가장 많이 사용되는 캐시 교체 정책이다.

LFU(Least Frequently Used) 방식은 캐시된 콘텐츠 중 요청 빈도가 낮은 콘텐츠부터 저장 공간에서 삭제하고 새로운 콘텐츠를 저장하는 방법이다.

다음으로 FIFO(First In First Out) 방식은 캐시 내에서 가장 먼저 저장된 콘텐츠를 삭제하고 새로운 콘텐츠를 저장하는 방법으로 구현이 쉽지만, 특정 상황에서 너무 자주 교체되는 단점 때문에 효율성을 보장할 수 없다.

마지막으로 random cache 방식은 캐시 저장 공간에서 임의의 콘텐츠를 선택하여 삭제하고 새로운 콘텐츠를 저장하는 방식이다<sup>12,13)</sup>.

### 2.2 Mobile Edge Computing

Mobile Edge Computing 기술은 5G 이동통신의 핵심 기술로 모바일 네트워크 엣지에서 cloud computing, IT 서비스 환경을 제공한다. Application 개발자 및 콘텐츠 provider들에게 실시간 서비스를 제공하기 위한 ultra low latency와 high bandwidth를 제공하며 최근 이에 관한 연구가 진행되고 있다.

Orsini의 연구<sup>14)</sup>에서는 기존 Mobile Cloud Computing의 high latency, low bandwidth 때문에 cloud gaming과 같은 실시간 서비스 제공이 어렵다는 점에서 mobile application 개발자를 위해 Mobile Edge Computing 개념을 활용한 프레임워크를 제안하였다. 또한, Nunna의 연구<sup>15)</sup>에서는 Mobile Edge Computing의 개념을 포함한 구체적인 5G 이동통신 네트워크의 구조 및 제공 가능한 실시간 서비스에 대해 제안하였다.

Mobile Edge Computing의 표준화를 진행하는 ETSI(European Telecommunications Standards

Institute)에서 제안하고 있는 use case로는 비디오 분석, Internet of Things(IoT), 증강 현실, 최적화된 로컬 콘텐츠 분배 및 데이터 캐싱 등이 있으며 데이터 캐싱의 경우, edge server가 코어망과 소형셀 사이에 연결되며 여러 소형셀과 연결되어 소형셀 사용자의 데이터 트래픽을 캐싱한다. Mobile Edge Computing의 개념을 활용하여 edge server에 캐시를 할 경우, 소형셀 백홀 링크의 사용량을 최대 35%까지 줄일 수 있다는 장점이 있다<sup>[6]</sup>.

### 2.3 소형셀 환경

소형셀 포럼에서는 그림 1과 같이 소형셀의 범위 및 용도에 따라 크게 home, enterprise, urban, rural과 같이 4가지로 분류하였다. Home 소형셀은 집 또는 작은 사무실에 설치되는 소형셀이고, enterprise 소형셀은 회사, 소형 사무실 등에 설치되어 home 소형셀 대비 넓은 커버리지를 갖는다. Urban 소형셀은 공항, 경기장, 놀이동산 등에 설치되며 rural 소형셀은 일시적인 재난지역 혹은 배, 비행기 등에 설치되는 소형셀이다<sup>[7]</sup>.

위 4가지 소형셀 중 enterprise, urban 소형셀 환경은 home 소형셀 대비 넓은 범위를 커버하며 여러 사용자가 접속할 수 있다. 따라서 효과적인 캐시를 위해 소형셀 사용자의 특성을 파악하여 사용자가 요청할 것으로 예측되는 콘텐츠를 캐시 저장 공간에 미리 저장할 필요가 있다.

소형셀 캐시 관점에서 enterprise/urban 소형셀의 특징을 보면 enterprise 소형셀의 경우, 위에서 설명한 것처럼 소규모 스토어, 회사 사무실 같은 작은 범위를 서비스하기 때문에 개인의 취향과 특징이 비슷한 사용자들끼리 같은 소형셀에 접속할 확률이 높다. 예를 들어 대학 연구실에 설치된 소형셀에 접속한 사용자는 대학원생일 확률이 높고, 대학원생의 특성에 따라 요청할 콘텐츠를 예측할 수 있다. 다음으로 urban 소형셀 환경의 경우, 배치되는 장소가 대부분 특정한 목적을 가지고 있는 경우라 장소에 맞게 동일한 목적을

가진 사용자들이 모일 확률이 높다. 예를 들어 야구장에서는 그 날 경기 중인 해당 구단 및 선수의 정보를 검색할 것이다. 따라서 캐시를 해야 할 콘텐츠 및 주제가 정해져 있는 경우가 많아서 무엇을 캐시 할지 예측할 수 있다<sup>[11]</sup>.

### 2.4 소형셀 캐시 연구

코어망 트래픽 감소를 위한 한 방법인 소형셀 캐시 기법을 위해 최근 다음과 같은 연구가 진행되고 있다. 먼저 Shim의 논문<sup>[7]</sup>에서는 helper와 macro BS(Base Station)을 통해 사용자들이 트래픽을 처리 받는 femto-caching 기법에서 service provider가 인기 있는 video 콘텐츠를 예측하여 사용자들에게 가까운 helper에게 미리 배치하는 방법을 제안하였고, ElBamby의 논문<sup>[8]</sup>에서는 사용자가 요청한 콘텐츠 정보를 이용하여 사용자를 grouping하고 group 별로 같은 SCBS(Small Cell Base Station)에 접속하여 해당 SCBS에 접속한 group의 인기 콘텐츠를 저장하는 방법을 제안하였다. 또한, Su의 논문<sup>[9]</sup>에서는 D2D 통신이 가능한 CDN(Content Centric Network)환경에서 사용자의 social network 정보를 활용한 캐시 방법을 제안하였다. 소형셀 기지국에 위치한 사용자가 요청한 콘텐츠를 캐시할 때 콘텐츠를 제공받기 위한 delay를 cost로 정하고 cost가 낮은 콘텐츠를 우선적으로 캐시하도록 하였고, Xing의 논문<sup>[10]</sup>에서는 mobile social network 환경에서 사용자 간의 social relationship을 활용하여 cache block model을 제안하였다. 제안한 cache block model은 사용자간의 social 연결 상태, 콘텐츠의 크기 등을 고려한 model로 캐시 적중률 측면에서 기존 LRU 방식에 비해 약 10%이상의 성능 향상을 확인하였다. Jung의 논문<sup>[11]</sup>에서는 home 소형셀 환경에서 적용할 수 있는 사용자 컨텍스트 기반 캐시 기법을 제안하였다. 하나의 소형셀 사용자의 특성이 유사하다는 점을 이용하여 사용자 특성에 따른 인기 콘텐츠를 예측하여 캐시하도록 하였다.

기존 연구<sup>[6-10]</sup>에서는 소형셀 사용자의 특성을 반영하지 않고, 일반적인 인기 콘텐츠의 효율적 배치만을 고려하고 있으며, D2D 환경에서 사용자의 social network 정보를 활용한다는 점에서 D2D 통신이 가능하지 않은 환경 또는 사용자의 social network 정보를 활용할 수 없을 경우 적용할 수 없다는 단점이 있다. 또한, 사용자의 특성을 반영한 Jung의 연구<sup>[11]</sup>에서는 사용자 특성이 유사한 home 소형셀 환경만을 가정하고 있어서 넓은 범위에 설치되어 많은 사용자를 서비스하는 enterprise/urban 소형셀 환경에서는 적용하기

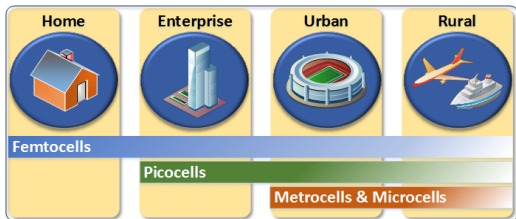


그림 1. 사용범위 및 용도에 따른 소형셀 분류  
Fig. 1. Classification according to the coverage and purpose

어렵다는 단점이 있다.

### III. 제안하는 소형셀 캐시 알고리즘

#### 3.1 소형셀 캐시 알고리즘을 위한 네트워크 구조

제안하는 소형셀 캐시 알고리즘이 적용될 경우 예상되는 네트워크 구조는 그림 2와 같다. 소형셀에 존재하는 모든 사용자는 사용자 특성 및 사용자가 접속한 소형셀의 위치정보에 따라 그룹화된다. 모든 사용자는 소형셀 기지국을 통하여 콘텐츠를 다운로드 받으며 각 소형셀 기지국은 edge server를 거쳐 코어망과 연결된다. Edge server에 위치한 캐시는 소형셀에 존재하는 각 그룹 별로 저장 용량의 크기를 다르게 할당하여 그룹별 인기 콘텐츠를 저장한다. 사용자가 캐시에 저장된 콘텐츠를 요청할 경우 해당 콘텐츠는 코어망을 통하지 않고 캐시를 통해 바로 사용자에게 전송되고 그렇지 않을 경우 코어망을 통해서 사용자에게 제공된다.

#### 3.2 제안하는 소형셀 캐시 알고리즘

제안하는 소형셀 캐시 알고리즘은 아래와 같이 4단계로 구분할 수 있다. 그림 3과 같이 pseudo code로 나타낼 수 있으며 사용된 파라미터에 대한 설명은 표 1과 같다.

##### 3.2.1 Phase 1 : User grouping

소형셀 사용자 그룹을 싱글녀, 싱글녀, 직장인, 재테크족, 주부, 대학생, 청소년 중 사용자 특성과 가장 유사한 그룹 하나를 선정한다<sup>18)</sup>. 소형셀에 접속한 사용자는 자신의 특성을 직접 선택하거나 웹 로그 분석 및 과거 검색기록 분석을 통해 알 수 있다. 또한, edge server에 의해 사용자가 현재 접속한 소형셀에 따라 사용자 그룹이 선택되어 각 사용자 그룹  $G_i$ 로 그룹화

되며  $i$ 는 각 사용자 그룹을 나타내는 인덱스이다.

##### 3.2.2 Phase 2 : Initial caching

Edge server에 위치한 캐시 저장 공간은 소형셀에 존재하는 사용자 그룹별로 동일한 용량이 할당

되며 각 사용자 그룹의 캐시 용량은 아래의 수식 (1)과 같이 정의된다.

$$S_i = S/n. \tag{1}$$

위 수식 (1)에서  $S_i$ 는 사용자 그룹  $G_i$ 에게 할당되는 캐시 저장 공간 Cache( $i$ )의 크기,  $S$ 는 edge server에 위치한 전체 캐시 저장 용량,  $n$ 은 사용자 그룹의 수를 나타낸다. 할당된 그룹별 캐시 저장 공간 Cache ( $i$ )에는 각 그룹의 인기 콘텐츠를 예측하여 인기도 높은 콘텐츠부터 차례로 저장한다. 콘텐츠의 인기도는  $j$ 로 나타내며 낮은 값을 갖을수록 인기도가 높은 파일을 의미한다.

$$S_i \leq \sum_{j=1}^l F_{C_{ij}}. \tag{2}$$

위 수식 (2)에서  $C_{i,j}$ 는 사용자 그룹  $G_i$ 의  $j$ 번째 인기 콘텐츠,  $F_{C_{ij}}$ 는 해당 파일의 크기를 나타내며 할당된 저장 용량의 크기  $S_i$ 를 넘지 않을 때까지 인기 콘텐츠를 저장한다.

##### 3.2.3 Phase 3 : Cache storage size update

Edge server는 기준 시간 동안의 각 그룹 별 요청 트래픽량을 측정된 뒤 이에 비례하여 캐시 저장 공간을 그룹별로 다시 할당한다. 먼저 사용자 그룹  $G_i$ 의 요청 트래픽량  $R_i$ 는 아래의 수식 (3)을 통해 구할 수 있으며 모든 사용자 그룹에 대해 수행한다.

$$R_i = \sum_{j=1}^l r_{i,j} \times F_{C_{ij}}, \tag{3}$$

$$S_i = \frac{R_i}{\sum_{i=1}^n R_i} \times S. \tag{4}$$

위 수식(3)에서  $R_i$ ,  $r_{i,j}$ 는 그룹 별 요청 트래픽량, 사용자 그룹  $G_i$ 의 콘텐츠  $j$ 에 대한 요청 횟수를 나타낸다. 이를 통해 각 사용자 그룹  $G_i$ 의 요청 트래픽량을 계산한 뒤 수식 (4)와 같이 그룹 별 요청 트래픽량

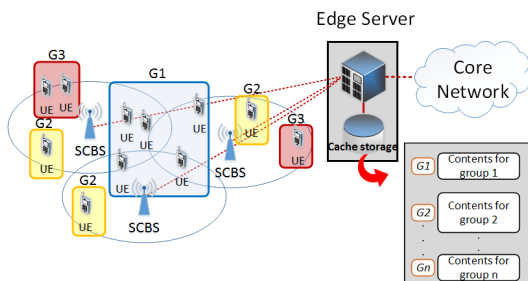


그림 2. 제안하는 소형셀 캐시 알고리즘을 위한 네트워크 구조  
Fig. 2. Network architecture for proposed small cell cache algorithm

에 비례하여 그룹별 캐시 용량을 다시 할당한다.

### 3.2.4 Phase 4 : Contents update

캐시 저장 공간에 저장된 그룹별 콘텐츠는 콘텐츠 업데이트 주기( $a_0$ )마다 새로운 인기 콘텐츠로 업데이트 된다. 하지만 사용자 그룹의 캐시 적중률이 일정  $threshold(H)$  이하로 낮아 질 경우, 신속히 새로운 콘텐츠를 저장하기 위해 콘텐츠 업데이트 주기를 초기 콘텐츠 업데이트 주기( $a_0$ )에서  $p$  시간만큼 감소시킨다.  $a_0 - p$  시간이 지난 후 다시 캐시 적중률을 확인하여 이하로 낮을 경우 다시  $p$  시간만큼 감소시키며 캐시 적중률이  $H$  이상이 될 경우 콘텐츠 업데이트 주기는 초기값( $a_0$ )으로 변경한다. 콘텐츠 업데이트 수행을 위해 기준 시간 동안 캐시에 저장된 콘텐츠 중 요청량이 0인 콘텐츠를 제거하고, 빈 저장 공간에 새로운 인기 콘텐츠를 저장한다. 아래의 수식 (5)에 따라 제거된 콘텐츠의 크기를 넘지 않을 때까지 해당 시간의 인기 콘텐츠가 저장 된다.

$$F_{C_{i,j,0}} \leq \sum_{j=1}^l F_{C_{i,j}} . \quad (5)$$

위 수식(5)에서  $F_{C_{i,j,0}}$ 는 캐시에 저장된 파일 중 기준 시간 동안 요청량이 0인 파일의 크기를 나타내며 해당 파일을  $Cache(i)$ 에서 삭제 한 뒤 새로운 콘텐츠  $C_{i,j}$ 를 빈 공간에 저장한다. 단, 사용자 그룹의 요청 트래픽량과 콘텐츠 업데이트를 위한 트래픽량을 비교한 뒤 업데이트 수행 여부를 결정한다.

$$R_i = \sum_{j=1}^l r_{i,j} \times F_{C_{i,j}} \geq B_i . \quad (6)$$

위 수식 (6)에서  $B_i$ 는 콘텐츠 업데이트 수행을 위해 코어망을 통해 전송되는 트래픽을 의미하며, 사용자 그룹의 요청 트래픽량이 콘텐츠 업데이트를 위한 트래픽량보다 작을 경우 콘텐츠 업데이트를 수행하지 않도록 하였다. 업데이트를 수행함으로써 사용자 그룹의 콘텐츠 요청량에 비해 더 큰 트래픽을 발생시켜 캐시 효율을 감소시킬 수 있기 때문이다.

표 1. 제안하는 소형셀 캐시 알고리즘 파라미터  
Table 1. Parameters of proposed small cell cache algorithm

Parameters	Description
$S$	Total cache storage size
$Cache(i)$	Cache storage of group $G_i$

$S_i$	Size of cache storage allocated group $G_i$
$i$	Index of group $G_i$
$j$	Popularity rank of contents
$C_{i,j}$	Popular contents of group $G_i$
$F_{C_{i,j}}$	Size of $C_{i,j}$
$C_{i,j,0}$	Cached contents of which number of request is 0
$F_{C_{i,j,0}}$	Size of $C_{i,j,0}$
$r_{i,j}$	Number of request for content $j$ of group $G_i$
$t_0$	Start time
$t_{final}$	Finish time
$a_0$	Initial frequency of the contents update
$B_i$	Traffic size for the contents update of group $G_i$
$H_{i,t}$	Cache hit ratio of group $G_i$ at time $t$

#### Phase 1: User grouping

Classify users into groups  $G_i$  based on user's characteristics for  $i = 1, 2, \dots, n$

#### Phase 2: Initial caching

At initial time  $t_0$ : Allocate same cache storage size for each group

$$S_i = S/n \text{ for } i = 1, 2, \dots, n$$

For  $j = 1 : l$

Pre-fetch  $C_{i,j}$  at  $Cache(i)$  for  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, l$

$$\text{If } S_i \leq \sum_{j=1}^l F_{C_{i,j}}$$

Break

End if

End

#### Phase 3: Update cache storage size

For  $t = t_0 : a_0 : t_{final}$

$$\text{Calculate } R_i = \sum_{j=1}^l r_{i,j} \times F_{C_{i,j}} \text{ for } i = 1, 2, \dots, n$$

$$S_i = \frac{R_i}{\sum_{i=1}^n R_i} \times S \text{ for } i = 1, 2, \dots, n$$

End

#### Phase 4: Update fetched contents

For  $t = t_0 : a : t_{final}$

If  $H_{i,t} \geq H$

$$a \leftarrow a_0$$

Remove  $C_{i,j,0}$  at  $Cache(i)$

For  $j = 1 : l$

```

Pre-fetch  $C_{i,j}$  at Cache( $i$ ) for  $i = 1, 2, \dots, n$ 
and  $j = 1, 2, \dots, l$ 
If  $F_{C_{i,0}} \leq \sum_{j=1}^l F_{C_{i,j}}$ 
    Break
End if
End

Else
 $a \leftarrow a - p$ 
Remove  $C_{i,j,0}$  at Cache( $i$ )
For  $j = 1 : 1 : l$ 
    Calculate  $\sum_{j=1}^l F_{C_{i,j}}$ 
    If  $F_{C_{i,0}} \leq \sum_{j=1}^l F_{C_{i,j}}$ 
         $B_i \leftarrow \sum_{j=1}^l F_{C_{i,j}}$ 
        Break
    End if
End
End
If  $\sum_{j=1}^l r_{i,j} \times F_{C_{i,j}} \geq B_i$ 
    For  $j = 1 : 1 : l$ 
        Pre-fetch  $C_{i,j}$  at Cache( $i$ ) for  $i = 1, 2, \dots, n$ 
        and  $j = 1, 2, \dots, l$ 
        If  $F_{C_{i,0}} \leq \sum_{j=1}^l F_{C_{i,j}}$ 
            Break
        End if
    End
End if
End if
End

```

그림 3. 제안하는 소형셀 캐시 알고리즘의 의사코드  
 Fig. 3. pseudo code of proposed small cell cache algorithm

#### IV. 성능 분석

##### 4.1 성능 지표

제안하는 소형셀 캐시 알고리즘의 성능 평가를 위한 성능지표로 cache efficiency를 사용한다<sup>[11]</sup>. Cache efficiency는 성능 분석을 수행하는 동안 발생한 전체 트래픽량 중 캐시에 저장된 콘텐츠를 통해 제공받은 트래픽량의 비율로 아래의 수식과 같이 표현할 수 있다.

$$Cache\ efficiency = \frac{T_{total} - (T_{Save} + T_{Cache\ miss})}{T_{total}} \times 100. \quad (7)$$

표 2. 성능분석 파라미터  
 Table 2. Parameters of the performance evaluation

Parameters	Values
Type of smallcells	Enterprise/urban small cells
Number of smallcells	4
Number of users' groups	4
Size of a content	7.8 MB
Probability to search a keywords	1~10 ranked keywords : 56.64% 11~100 ranked keywords : 43.36%
Average number of requests of each users	59.6 times
Initial frequency for cache update ( $\alpha_0$ )	3 hours
Decrement value of cache update frequency ( $p$ )	1 hours
Size of total cache storage equipped at edge server	416 MB

위 수식 (7)의  $T_{total}$ 는 성능 분석 기간 동안 발생한 전체 트래픽량,  $T_{Save}$ 는 캐시에 그룹별 인기 콘텐츠를 저장하기 위해 발생하는 트래픽량,  $T_{Cache\ miss}$ 는 캐시 미스에 의해 코어망을 통해 전송되는 트래픽량을 의미한다.

##### 4.2 성능 분석 환경

제안하는 소형셀 캐시 알고리즘의 성능 평가를 위한 파라미터는 아래의 표 2<sup>[11]</sup>와 같다. 성능 분석을 위한 수단으로 국내 주요 검색 포털 사이트 중 전체 시장의 85.94%의 점유율(2016년 04월 22일 기준)을 차지하는 네이버의 그룹별 인기 검색어를 활용하였다<sup>[18,19]</sup>. 싱글남, 싱글녀, 대학생, 청소년의 4개의 사용자 그룹에 대하여 2015년 12월 01일부터 2015년 12월 31일까지 총 1개월 동안의 인기 검색어를 수집하였고, 각 그룹의 사용자 수는 20명으로 설정하였다. 소형셀 캐시 저장 용량은 일반적인 웹 브라우저의 평균 캐시 저장 용량인 104 Mbyte를 사용하였고 총 4개의 소형셀이 존재하는 환경을 가정하였다. 소형셀 내부에 캐시가 위치한 경우의 캐시효율과 비교하기 위해 edge server의 캐시 저장 용량은 소형셀 캐시 용량의 4배인 416 Mbyte로 하였다.

성능 분석은 기존에 Jung의 논문<sup>[11]</sup>에서 제안된 소형셀 캐시 알고리즘과 본 논문에서 제안한 알고리즘의 캐시 적중률, cache efficiency를 비교하였다. 기존 알고리즘에서는 캐시 저장 공간이 edge server가 아닌 소형셀에 위치하며 저장 공간의 크기는 104 MByte로

일정하다. 또한, 캐시 업데이트 주기가 일정하다. 제안한 알고리즘은 그룹의 캐시 저장 용량의 크기가 동일할 때와 그룹의 콘텐츠 요청량에 비례하여 저장 용량이 다를 경우에 대해 캐시 적중률 threshold 값( $H$ )의 변화에 따른 캐시 적중률과 cache efficiency를 비교하였다.

### 4.3 성능 분석 결과

#### 4.3.1 콘텐츠 업데이트 주기 변화에 따른 사용자 그룹별 캐시 적중률

그림 4는 각 사용자 그룹의 캐시 저장 용량의 크기를 동일하게 할 경우(각 그룹의 저장용량=104 Mbyte),  $H$ 값의 변화에 따른 각 사용자 그룹의 캐시 적중률을 나타낸다.  $H$ 값이 커질수록 콘텐츠 업데이트 주기가 짧아지게 되고, 이 때문에 모든 사용자 그룹의 캐시 적중률이 증가함을 확인 할 수 있다. 또한, 분석 결과 싱글남, 싱글녀 그룹의 캐시 적중률이 대학생, 청소년 그룹의 캐시 적중률보다 낮은 것을 확인 할 수 있다. 이는 싱글남, 싱글녀 그룹의 사용자가 대학생, 청소년 그룹에 비해 다양한 사용자가 포함될 수 있으며 그로 인해 인기 검색어의 종류가 다양해져 인기 검색어의 매시간 변화가 대학생, 청소년 그룹에 비해 싱글남, 싱글녀 그룹의 변화가 크기 때문으로 생각할 수 있다.

그림 5는 캐시 저장 용량을 각 그룹별로 다르게 할 당하였을 경우,  $H$ 값의 변화에 따른 각 사용자 그룹의 캐시 적중률을 나타낸다. 캐시 저장 용량은 기준 시간 동안의 사용자 그룹  $G_i$ 의 콘텐츠 요청 횟수, 요청하는

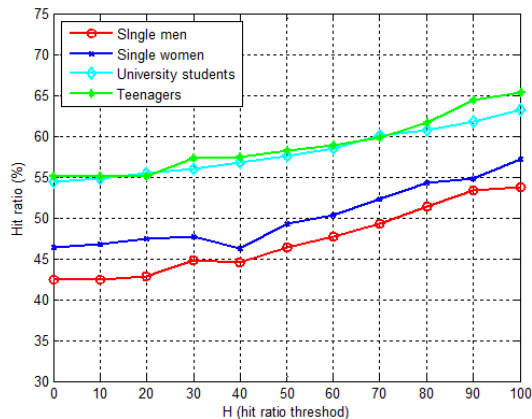


그림 4. 그룹별 캐시 용량이 일정 할 때의 그룹별 캐시 적중률  
Fig. 4. Cache hit ratio of each group in case of same storage size for each group

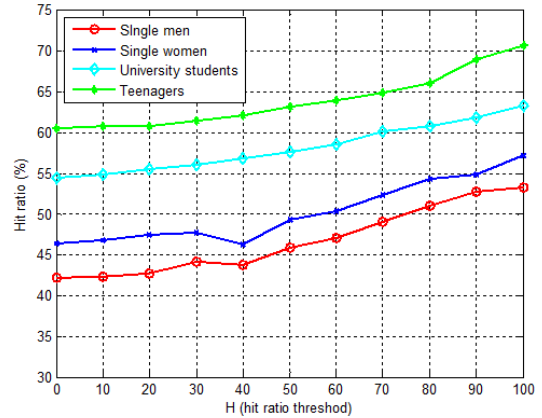


그림 5. 그룹별 캐시 용량이 다를 때의 그룹별 캐시 적중률  
Fig. 5. Cache hit ratio of each group in case of different storage size for each group

콘텐츠 사이즈에 비례하며 싱글남, 싱글녀, 대학생, 청소년 그룹의 사용자 수를 10, 20, 20, 30명으로 변경하여 그룹별 캐시 저장 용량이 다르게 할당되도록 하였다. 그림 4의 결과와 같이  $H$ 값이 증가함에 따라 콘텐츠 주기가 감소하면서 모든 그룹의 캐시 적중률이 증가하는 것을 확인할 수 있다. 또한, 분석 결과 다른 그룹에 비해 큰 캐시 저장 용량을 할당받은 대학생 그룹의 경우 그림 4의 결과보다 캐시 적중률이 높아지는 것을 확인할 수 있다. 이는 저장 용량이 커짐에 따라 캐싱할 수 있는 콘텐츠가 많아지기 때문이다.

#### 4.3.2 콘텐츠 업데이트 주기 변화에 따른 평균 캐시 적중률

그림 6은 제안한 알고리즘과 기존에 제안된 알고리즘을 적용하였을 때의 평균 캐시 적중률을 나타낸다. 기존에 제안된 알고리즘은 업데이트 주기를 변경하지

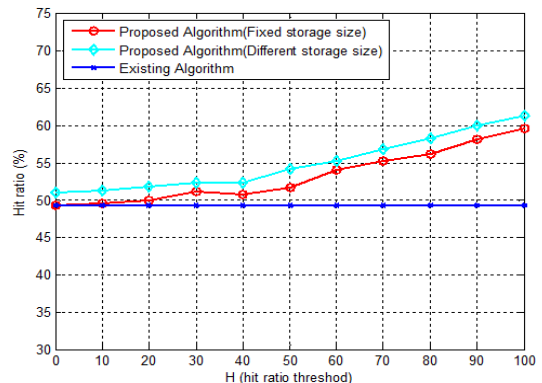


그림 6. 그룹별 평균 캐시 적중률  
Fig. 6. Average cache hit ratio of each group

않기 때문에 항상 일정한 값을 갖는 것을 확인할 수 있다. 또한, 그룹별 캐시 저장 용량을 다르게 할당할 경우 일정한 용량을 할당할 경우보다 높은 캐시 적중률을 얻을 수 있었다. 이는 그룹별 캐시 용량을 달리 하였을 때, 큰 저장 용량을 할당 받은 청소년 그룹의 캐시 적중률의 증가는 큰 반면, 작은 용량을 할당받은 싱글남 그룹의 캐시 적중률 감소는 그에 비해 크지 않기 때문이다. 따라서 제안한 알고리즘을 적용한 경우, 기존 알고리즘 대비 최대 11% 높은 캐시 적중률을 얻을 수 있었다.

#### 4.3.3 콘텐츠 업데이트 주기 변화에 따른 평균 cache efficiency

그림 7은 캐시를 사용함으로써 코어망 트래픽 감소 효과를 확인할 수 있는 cache efficiency를 나타낸다. 분석 결과 기존의 알고리즘 대비 제안한 알고리즘의 cache efficiency가 높게 나타남을 확인할 수 있다. 이는 캐시 적중률 향상으로 인해 코어망을 통해 전송되는 트래픽이 감소하였기 때문이다. 제안한 알고리즘에서  $H$ 값이 0일 경우, 사용자의 요청 트래픽량에 비례하여 그룹별 캐시 용량을 다르게 함으로써 cache efficiency가 증가함을 확인할 수 있다.

또한,  $H$ 값이 증가함에 따라 캐시 적중률이 증가한 것과 다르게 cache efficiency의 경우,  $H$ 값이 60일 때 최대 cache efficiency 값을 갖고,  $H$ 값이 70 이상 일 때부터는 오히려 cache efficiency가 감소함을 확인할 수 있다. 이는  $H$ 값이 커짐에 따라 잦은 캐시 업데이트 수행으로써 코어망을 통해 전송되는 트래픽이 증가하였기 때문이다. 이를 통해 사용자의 요청 트래픽량에 비례하여 그룹별 캐시 저장 용량을 다르게 할당하고, 적절한  $H$ 값을 선택할 경우 캐시 효율 측면에서

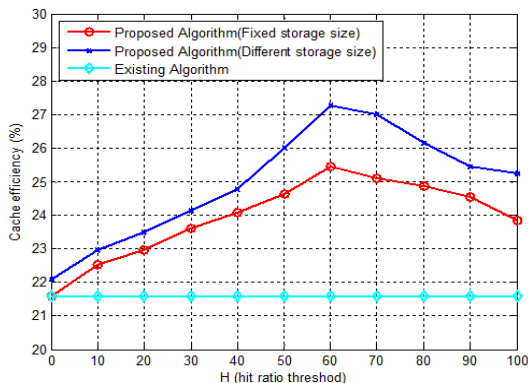


그림 7. 그룹별 평균 캐시 효율  
Fig. 7. Average cache efficiency of each group

더 좋은 성능을 보임을 확인하였다.

## V. 결 론

본 논문에서는 home 소형셀 환경과 달리 다양한 특성을 지닌 사용자를 서비스하고 넓은 지역을 커버하는 enterprise/urban 소형셀 환경에서 적용 가능한 캐시 알고리즘을 제안하였다.

제안한 알고리즘에서 캐시 저장 공간은 소형셀 기지국이 아닌 edge server에 위치하도록 하였고, 사용자들의 다양한 특성을 반영하기 위해 소형셀 사용자를 사용자 특성 및 사용자가 위치한 소형셀에 따라 그룹화하여 그룹별 인기 콘텐츠를 예측하여 저장하는 방법을 사용하였다. 캐시 적중률에 따라 콘텐츠 업데이트 주기를 다르게 하고, 각 그룹의 요청 트래픽량에 비례하여 캐시 저장 용량을 다르게 할당하여 기존 캐시 알고리즘 대비 코어망 트래픽 감소 효과를 증대시켰다. 성능 분석 결과 기존 알고리즘 대비 캐시 적중률 측면에서 약 11%, cache efficiency 측면에서 약 5.5%의 성능 향상을 확인할 수 있었다.

본 논문에서 제안한 알고리즘은 향후 넓은 범위를 커버하고, 다양한 특성을 지닌 사용자가 존재하는 enterprise/urban 소형셀 환경에 적용 가능할 것으로 예상되며 폭증하는 모바일 트래픽으로 인한 코어 네트워크의 트래픽 감소에 도움이 될 것으로 기대된다.

## References

- [1] Y. K. Kim, S. H. Lee, and Y. J. Kim, "Trends on 5G communications," *ETRI Electron. and Telecommun. Trends*, vol. 30, no. 1, pp. 1-11, Feb. 2015.
- [2] Samsung Electronics, *5G Vision White Paper*, 2015.
- [3] Cisco white paper, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014 -2019*, Feb. 2015.
- [4] N. H. Sung, Y. J. Choi, and J. W. Jang, "Joint operation of ABS with power control and derivation of an effective ABS ratio for LTE hetnet environments," *J. KICS*, vol. 40, no. 12, pp. 2381-2388, Dec. 2015.
- [5] Y. Jang, E. Cho, and E. Hong, "Effect of interference mitigation technique and performance analysis for small cell in



homogeneous networks,” *J. KICS*, vol. 39C, no. 10, pp. 937-945, Oct. 2014.

[6] T. Wang, L. Song, and Z. Han, “Dynamic femtocaching for mobile users,” in *Proc. WCNC 2015*, pp. 861-865, New Orleans, LA, Mar. 2015.

[7] J. N. Shim, B. Y. Min, K. Kim, and D. K. King “Advanced femto-caching file placement technique for overlapped helper coverage,” in *Proc. VTC 2014 Spring*, May 2014.

[8] M. S. ElBamby, M. Bennis, and M. Latva-aho, “Content-aware user clustering and caching in wireless small cell networks,” in *Proc. ISWCS 2014*, pp. 945-949, Barcelona, Spain, Aug. 2014.

[9] Z. Su and Q. Xu, “Contents distribution over content centric mobile social networks in 5G,” *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 66-72, Jun. 2015.

[10] Q. Xing, Y. Li, J. Wang, and Y. Han, “A user-relationship-based cache replacement strategy for mobile social network,” in *Proc. 2015 Int. Conf. Frontier of Comput. Sci. and Technol.*, Mar. 2015.

[11] S. Y. Jung and J. H. Kim, “Caching algorithm for core network offloading in smallcell environment,” *J. IEIE*, vol. 52, Mar. 2015.

[12] S. Han, H. Park, and T. Kwon, “Shelf-life time based cache replacement policy suitable for web environment,” *J. KICS*, vol. 40, no. 6, pp. 1091-1101, Jun. 2016.

[13] A. Balamash and M. Krunz, “An overview of web caching replacement algorithms,” *IEEE Commun. Surveys & Tuts.*, pp. 44-56, 2004.

[14] G. Orsini, D. Bade, and W. Lamersdorf, “Computing at the mobile edge: Designing elastic android applications for computation offloading,” in *Proc. WMNC 2015*, pp. 112-119, Munich, Germany, Oct. 2015.

[15] S. Nunna, A. Kousaridas, M. Ibrahim, and M. Dillinger, “Enabling real-time context aware collaboration through 5G and mobile edge computing,” in *Proc. ITNG 2015*, pp. 601-605, Las Vegas, NV, Apr. 2015.

[16] ETSI, *Mobile-edge computing—introductory*

*technical white paper*, Sept. 2014.

[17] Small Cell Forum Release 1 Document 046.01.01, *Small Cell Services*, Feb. 2013.

[18] Naver, <https://naver.com>

[19] INTERNET TREND, <http://internettrend.co.kr/trendForward.tsp>

정 현 기 (Hyun Ki Jung)



2014년 : 아주대학교 전자공학과 졸업  
 2014년~현재 : 아주대학교 전자공학과 석사과정  
 <관심분야> 소형셀, 무선 캐시, 차세대 이동통신 시스템 등

정 소 이 (Soyi Jung)



2012년 : 아주대학교 전자공학과 졸업  
 2014년 : 아주대학교 전자공학과 석사과정 졸업  
 2016년~현재 : 아주대학교 전자공학과 박사과정

<관심분야> 소형셀, LTE/LTE-A, 차세대 이동통신 시스템 등

이 동 학 (Dong Hak Lee)



2016년 : 아주대학교 전자공학과 졸업  
 2016년~현재 : 아주대학교 전자공학과 석사과정  
 <관심분야> 소형셀, 무선 캐시, 차세대 이동통신 시스템 등

이 승 규 (Seung Que Lee)



1988년 : 중앙대학교 전산학과 졸업  
1990년 : 중앙대학교 전산학과 석사과정 졸업  
2007년 : 고려대학교 전산학과 박사과정 졸업  
1990년~현재 : 한국전자통신연구원 책임연구원

<관심분야> 5세대 이동통신 시스템, 소형셀 시스템, 서비스 플랫폼, 네트워크 기능 가상화, 모바일 엣지 컴퓨팅 등

김 재 현 (Jae-Hyum Kim)



1987년~1996년 : 양대학교 전산과 학사 및 석/박사 졸업  
1997년~1998년 : 미국 UCLA 전기전자과 박사 후 연수  
1998년~2003년 : Bell Labs, NJ, USA, 연구원  
2003년~현재 : 아주대학교 전자공학부 교수

<관심분야> QoS/QoS, 무선 MAC 프로토콜, IEEE 802.11/15, B5G 통신 시스템, 국방 기술네트워크, 위성시스템 등