

Edge Caching Strategy with User Mobility in Heterogeneous Cellular Network Environments

Yoonjeong Choi[†] · Yujin Lim^{††}

ABSTRACT

As the use of mobile data increases, the proportion of video content is increasing steeply. In order to solve problems that arise when mobile users receive data from geographically remote cloud servers, methods of caching data in advance to edge servers geographically close to the users are attracting lots of attention. In this paper, we present a caching policy that stores data on Small Cell Base Station(SBS) to effectively provide content files to mobile users by applying a delayed offloading scheme in a cellular network. The goal of the proposed policy is to minimize the size of data transmitted from Macro Base Station(MBS) because the delayed offloading scheme requires more cost than when downloaded from MBS than from SBS. The caching policy is proposed to determine the size of content file and which content file to be cached to SBS using the probability of mobile users' paths and the popularity of content files, and to replace content files in consideration of the overlapping coverage of SBS. In addition, through performance evaluation, it has been proven that the proposed policy reduces the size of data downloaded from MBS compared to other algorithms.

Keywords : Edge Caching, Proactive Caching, User Mobility, Cellular Network

이종 셀룰러 네트워크 환경에서 사용자 이동성을 고려한 엣지 캐싱 기법

최 윤 정[†] · 임 유 진^{††}

요 약

모바일 데이터의 사용이 늘어나면서 특히 비디오 콘텐츠가 차지하는 비중이 가파르게 증가하고 있다. 모바일 사용자가 지리적으로 원거리에 위치한 클라우드 서버를 통해 데이터를 전달받으면서 발생하는 문제들을 해결하기 위해 사용자와 지리적으로 가까운 엣지 서버에 미리 데이터를 캐싱하는 방법이 많은 주목을 받고 있다. 본 논문에서는 셀룰러 네트워크 환경에서 지연 오프로딩 스킴(delayed offloading scheme)을 적용해 모바일 사용자에게 효과적으로 콘텐츠 파일을 제공하기 위한 SBS 캐싱 기법을 제안하였다. 지연 오프로딩 스킴에서 Macro Base Station(MBS)보다 Small Cell Base Station(SBS)으로부터 데이터를 다운받는 경우 더 적은 비용을 요구하기 때문에 MBS로부터 전송받는 데이터 크기를 최소화하는 것을 목표로 하였다. 모바일 사용자의 이동 경로 확률과 콘텐츠 파일의 인기도를 사용해 SBS에 캐싱할 콘텐츠 파일과 그 크기를 결정하고 SBS의 서비스 범위가 중복되는 것을 고려해 콘텐츠 파일을 재배치하는 캐싱 기법을 제안하였다. 또한 실험을 통해 다른 알고리즘보다 MBS로부터 다운받는 데이터 크기를 줄일 수 있다는 것을 증명하였다.

키워드 : 엣지 캐싱, 사전 캐싱, 사용자 이동성, 셀룰러 네트워크

1. 서 론

전 세계적으로 모바일 디바이스의 사용이 늘어나고 있으며 이에 따라 모바일 데이터 트래픽도 함께 증가하는 추세이다.

모바일 데이터 트래픽은 2017년부터 2022년까지 46% 정도의 연평균 성장률을 보일 것으로 예측하고 있다. 또한 2022년에 예상되는 모바일 데이터 트래픽은 77EB/month인데, 이 중 비디오로 인한 트래픽이 61EB/month를 차지할 정도로 비디오에 대한 수요가 매우 큰 것을 알 수 있다[1]. 따라서 많은 비디오 콘텐츠에 대한 수요를 감당하기 위해서 짧은 지연 시간을 제공할 수 있는 능력과 매우 많은 양의 데이터를 처리하는 능력이 중요해졌다. 하지만 클라우드를 통해 데이터를 전달받을 경우 사용자로부터 멀리 있기 때문에 지연 시간이 길고 클라우드에 많은 트래픽이 몰려 네트워크에 혼잡성이 가중된다[2].

※ 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2021R1F1A1047113).
※ 이 논문은 2021년 한국정보처리학회 ACK 2021의 우수논문으로 "셀룰러 네트워크에서 딜레이드 오프로딩 스킴을 적용한 사용자 이동성 고려 캐싱 기법"의 제목으로 발표된 논문을 확장한 것임.
† 준 회원 : 숙명여자대학교 IT공학과 석사과정
†† 종신회원 : 숙명여자대학교 IT공학과 교수
Manuscript Received : December 23, 2021
Accepted : January 5, 2022
* Corresponding Author : Yujin Lim(yujin91@sookmyung.ac.kr)

이를 해결하기 위하여 사용자와 가까운 곳에 위치한 엣지(edge) 서버에 캐싱하는 기법이 생겨났다. 엣지 서버에 저장된 데이터를 사용자에게 전송해주면 클라우드에 집중된 데이터 트래픽을 완화시킬 수 있고 사용자와 거리가 상대적으로 근접하기 때문에 빠른 속도로 데이터 전송이 가능하다[3]. 또한 하나의 클라우드에 많은 기기가 연결되면 대역폭이 부족해지면서 전송하는 패킷 손실의 위험이 있는데 이를 예방하고 전송 지연을 방지하는 효과를 가진다.

캐싱을 위한 엣지 서버는 용도에 따라 다양하다. SBS, MBS와 같은 셀룰러 네트워크에 캐싱을 할 수 있고, Road Side Unit(RSU)와 같이 와이파이를 이용해 접근이 가능한 곳에 캐싱하기도 한다. 또한 이동성이 있는 기기를 엣지 서버로 사용할 수도 있다. 예를 들어, 모바일 기기 혹은 이동 중인 자동차, 버스와 같은 운송 수단에 존재하는 저장 공간을 사용해 캐싱할 수도 있다.

모바일 사용자는 정적인 사용자와 다른 특성을 가지고 있는데 시간에 따라 데이터를 전송받는 위치가 계속 변경된다는 점이다. 사용자가 비디오 콘텐츠를 요구한 시간이나 장소에 따라 각각 다른 패턴을 보이기 때문에[4] 모바일 사용자의 요구를 만족시키기 위해서 시간에 따른 사용자의 이동성을 함께 고려하는 것이 좋다.

캐싱하는 환경을 구체적으로 설정하기 위해 지연 오프로딩 스킴(delayed offloading scheme)을 사용하였다. 지연 오프로딩 스킴을 MBS와 SBS로 이루어진 이중 네트워크에 적용하면 사용자가 요구한 콘텐츠를 정해진 시간 내에 SBS를 통해 모두 서비스받지 못했을 때 남은 데이터를 MBS로부터 전달받게 된다[5]. 이때 SBS보다 MBS로부터 다운받을 경우 필요한 비용이 더 커진다.

본 논문에서는 지연 오프로딩 스킴을 적용하여 MBS로부터 전송받는 데이터 양을 최소로 줄이는 것을 목표로 모바일 사용자의 이동성을 고려해 이중 셀룰러 네트워크에 콘텐츠를 캐싱하는 기법을 제안하였다. 실험을 통해 MBS로부터 다운받는 양을 다른 알고리즘에 비해 25~50% 정도 줄인 것을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서 움직이는 모바일 사용자를 위한 엣지 캐싱 기법들을 소개하였고 3장에서 새로운 알고리즘을 제시하였다. 4장에서는 기존의 다른 알고리즘과 비교를 통해 성능 평가를 하였고 5장에서 결론으로 마무리 지었다.

2. 관련 연구

이동성을 고려해 엣지 서버에 캐싱하는 방법은 모바일 디바이스나 차량과 같은 모바일 사용자의 종류, 캐싱 종류 그리고 캐싱 목적에 따라 모습이 다양하다. 먼저 모바일 사용자가 차량이라면 빠른 속도에 반응하기 위해 미리 캐싱을 해주는 기법을 사용하기도 하는데 이를 proactive caching이라고

한다. [6]에서는 차량의 이동 패턴을 고려해 데이터를 분산시켜 미리 캐싱하였다. RSU의 부담과 저장된 데이터의 중복을 줄이는 것을 목표로 하고 해당 RSU에 사용자가 요구하는 데이터가 없다면 주변의 가까운 RSU에서 데이터를 획득하는 방식을 사용하였다. [7]은 개인 정보를 보호하고자 연합학습(federated learning)을 사용해 콘텐츠의 인기를 예측하고, 예측된 콘텐츠 중 가장 인기 있는 데이터를 위주로 RSU에 미리 저장해두는 방식을 제안하였다.

데이터를 여러 곳에 저장하게 되면 저장된 장소가 서로 협력해 더 높은 hit ratio에 도달할 수 있고 짧은 지연 시간을 가지게 되는데 이를 cooperative caching이라고 한다[8]. [9]는 차량과 차량, 차량과 기지국이 서로 연결된 계층적 캐싱 구조를 5G 네트워크 환경에서 구성하였다. 사람, 차량 등 다양한 엔드 유저(end user)에게 서비스를 제공해 지연 시간을 감소시키고 캐싱할 수 있는 공간을 효율적으로 이용하였다.

데이터 캐싱을 결정할 때 불확실성이 높은 상황에 유연하게 대처하기 위해 계산된 확률에 의해 콘텐츠를 저장하는 것을 probabilistic caching이라고 한다[10]. [11]은 차량이 RSU 범위 안에 머무는 시간과 해당 범위 안에 동시에 존재하는 다른 차량을 고려해 확률적으로 RSU에 캐싱하는 기법을 제안하였다. [12]는 차량의 이동 경로를 예측하고, 예측 결과로 나온 확률에 따라 RSU에 콘텐츠 양을 달리해 캐싱하는 방법을 제안하였다.

캐싱 종류뿐 아니라 캐싱 목적에 따라서도 다양한 연구가 진행됐다. 사용자가 요구한 전체 데이터 중에서 전송받은 데이터의 비율을 hit ratio라 하는데 모바일 사용자가 모바일 기기일 경우 hit ratio를 높이기 위한 기법들이 제안됐다. [13]은 SBS와 헬퍼(helper)를 사용한 2단계 계층을 형성해 SBS의 범위에 속하는 헬퍼에는 해당 SBS에 캐싱된 콘텐츠와 다른 데이터를 저장해 hit ratio를 높였다. [14]는 hit ratio를 높이는 것을 목적으로 사용자 이동 패턴을 예측해 사용자 군집을 형성하고 분류된 군집에 알맞은 콘텐츠를 SBS에 캐싱하였다. 이를 통해 최종적으로는 에너지 효율을 높일 수 있게 되었다.

사용자에게 콘텐츠 전달 지연 시간을 최소화하는 것을 목표로 하는 연구도 있다. [15]는 기지국에 어떤 콘텐츠를 얼마나 많이 저장할지 결정하는 알고리즘을 통해 지연 시간을 줄이는 방법을 제안하였다. 모든 사용자의 평균적인 지연 시간을 최소화하기 위해 MATLAB의 linprog를 사용해 문제를 해결하였다.

[16]과 [5]는 셀룰러 네트워크에서 지연 오프로딩 스킴을 적용해 MBS로부터 다운받는 콘텐츠 양을 최소화하는 것을 목표로 하였다. 두 연구에서 모두 암호화된 캐싱(coded caching)을 다뤘고 [16]은 SBS가 매우 밀집되고 전송 속도가 낮은 환경에서 마르코프 체인을 통해 사용자의 움직임을 모델링하고 분산된 캐싱 알고리즘(distributed caching algorithm)을 제안하였다. [5]는 모바일 사용자의 경로 이동 확률과 콘텐츠의 인기를 사용해 MBS로부터 전송받는 데이터 양을 최소로 하

는 캐싱 기법을 고안하였다. 다만 SBS의 범위가 겹치는 것을 고려하지 않았는데 본 논문에서는 SBS가 밀도 높게 설치된 환경에서 SBS의 영역이 중첩되는 경우를 고려해 MBS로부터 평균적으로 전송받는 데이터 양을 최소화하는 캐싱 기법을 제안하였다.

3. 제안 기법

본 논문에서는 걷는 사람들이 지닌 모바일 디바이스를 모바일 사용자로 정의하였다. 이동하는 속도는 인간이 걷는 평균 속도를 고려해 1.25m/s로 결정하였다. 셀룰러 네트워크는 하나의 MBS와 다수의 SBS로 이루어진 이중 네트워크를 구성하고 각 콘텐츠 파일의 인기도와 이동하는 경로에 대한 확률은 사전에 주어진다 가정하였다.

본 연구의 목적은 모바일 사용자가 요구한 콘텐츠를 정해진 시간 내에서 SBS를 통해 최대한 많이 다운받아 MBS로부터 전송받는 데이터를 최소화하는 것이다. MBS로부터 콘텐츠를 전달받게 되면 SBS보다 많은 비용이 필요하기 때문이다.

MBS로부터 다운받는 데이터의 평균 크기를 계산하기 위해서는 각 경로 m 에서 SBS n 에 머무는 시간 $S_{m,n}$, SBS n 의 전송 속도 R_n , 구해야 하는 캐싱 정책 $x_{n,k}$ 이 필요하다. 본 논문에서는 모든 콘텐츠 파일의 크기가 B 로 같다고 가정하였고 각 경로 m 으로 이동할 때, MBS에서 다운받아야 하는 콘텐츠 k 의 크기 $d_{k,m}$ 을 식으로 나타내면 다음 Equation (1)과 같다[5].

$$d_{k,m} = \max\left\{B - \left(\sum_{n=1}^N \min\{x_{n,k}, R_n S_{m,n}\}\right), 0\right\} \quad (1)$$

알고리즘에서 사용하는 파라미터들은 Table 1에 정리하였다. MBS에서 다운받는 평균 데이터 크기를 구하기 위해서는 모든 가능한 경로와 파일의 인기도를 함께 고려해야 한다. 각 경로의 확률 q_m 와 파일의 인기도 p_k 를 고려해 SBS에서 전송받지 못해 MBS를 통해 전달되는 데이터의 평균 크기 d_{av} 는 다음 Equation (2)로 정리할 수 있다[5].

$$d_{av} = \sum_{m=1}^M \sum_{k=1}^K q_m p_k d_{k,m} \quad (2)$$

X_T 를 T 시간 이 주어졌을 때 결정되는 캐싱 방법이라고 정의하면, 위 식들을 이용해 MBS로부터 다운받는 데이터를 최소화하는 문제를 정의하면 다음 Equation (3), (4)와 같다.

$$\begin{aligned} P1: \min_{X_T} d_{av} \\ \text{subject to: } \sum_{k=1}^K x_{n,k} \leq C_n, \forall n. \end{aligned} \quad (3)$$

$$x_{n,k} \geq 0, \forall n, k. \quad (4)$$

Table 1. Parameters used in Proposed Algorithm

Symbols	Descriptions
$x_{n,k}$	the amount of content file k in SBS n
N	a set of SBSs, $n \in N$
C_n	cache capacity of SBS n
K	a set of content files, $k \in K$
R_n	transmission rate of SBS n
M	a set of mobile user paths, $m \in M$
$S_{m,n}$	resident time in SBS n when passing through the path m
q_m	the probability that a mobile user passes through the path m
p_k	popularity of content file k
B	size of a content file
T	total user mobility time
T_{\min}	the shortest time to download a requested content file using R_n

문제를 해결하기 위한 제안 알고리즘은 두 단계로 이루어져 있다. 첫 번째 단계는 SBS 영역이 겹치는 것을 고려하지 않고 캐싱을 하는 과정이고, 두 번째 단계는 SBS의 범위가 서로 겹치는 것을 고려해 캐싱된 파일을 재배치하는 과정이다. 전송 속도가 R_n 일 때, 콘텐츠 크기가 B 인 데이터를 모두 전송하기 위한 가장 짧은 시간을 T_{\min} 이라고 하면 T_{\min} 안에 SBS n 으로부터 받을 수 있는 파일 k 의 크기는 다음 Equation (5)와 같다[5].

$$\tilde{d}_{av,n}^k = \sum_{t=1}^T \sum_{m: S_{m,n} \geq t} p_k q_m \max\{\min\{x_{n,k} - (t-1)R_n, R_n\}, 0\} \quad (5)$$

지연 오프로딩 스킴을 적용했기 때문에 MBS에서 최소로 다운받는 것의 의미는 SBS를 통해 최대로 다운받는 것과 의미와 동일하다. 따라서 MBS로부터 전송을 적게 받기 위해서는 $\tilde{d}_{av,n}^k$ 을 최대로 만들면 된다. Equation (5)는 $\sum_{m: S_{m,n} \geq t} p_k q_m$ 가 커질수록 결과값이 같이 증가하므로 $\tilde{d}_{av,n}^k$ 를 최대로 만들기 위해서는 가장 큰 $\sum_{m: S_{m,n} \geq t} p_k q_m$ 를 선택해 Equation (5)에 대입하면 된다. 즉, 최대 $\sum_{m: S_{m,n} \geq t} p_k q_m$ 인 파일 k 를 R_n 만큼 SBS n 에 캐싱하면 된다. 하지만 이 방법은 T_{\min} 이 전체 시간 T 보다 작다면 최적의 값이 아닐 수도 있다. 따라서 추가적인 방법을 고려해 파일의 인기도에 더 큰 가중치 α 를 준 $\sum_{m: S_{m,n} \geq t} (p_k + \alpha)q_m$ 를 계산하였다. 인기도가 높은 파일 위주로 캐싱을 하게 되면 사용자가 요구한 콘텐츠의 전체 크기 B 를 모두 다운받는 것이 힘들어지기 때문에 β 만큼의 확률로 랜덤한 파일을 선택해 R_n 만큼 캐싱하고, $(1-\beta)$ 만큼의 확률로 가

장 큰 $\sum_{m: S_{m,n} \geq t} (p_k + \alpha) q_m$ 값을 가지는 콘텐츠 k 를 선택해 R_n 만큼 캐싱한다.

두 번째 단계는 SBS의 영역이 서로 중복되는 것을 고려하기 위해 각 SBS 범위가 다른 SBS와 중복되는 정도에 따라 범위를 나눠 저장된 파일을 재배포하는 것이다. 만약 모바일 사용자가 SBS들의 영역이 중복되는 위치에 있다면 각 SBS가 서로 다른 내용을 가지고 있을 때 다양한 파일을 저장하고 있어 모바일 사용자가 요구한 콘텐츠를 제공할 수 있는 가능성이 높아진다. SBS 사이의 영역 중첩 정도를 20% 이하, 20~40% 사이, 40% 이상으로 분류하고, 중복되는 범위가 클수록 해당 SBS는 다른 SBS와 협업할 수 있는 기회가 많아 다양한 콘텐츠를 제공할 수 있도록 재배포할 용량을 증가시켰다. 영역이 중복되지 않는 SBS에 대해서는 재배포를 고려하지 않았다. 두 SBS가 캐싱한 콘텐츠 중 동일한 파일을 저장한 용량의 특정 비율만큼 재배포했는데, 중첩되는 영역이 20% 이하면 τ , 20~40% 사이는 $(\tau \times 2)$, 40% 이상은 $(\tau \times 3)$ 만큼 재배포하였다.

재배포하기 위해 먼저 범위가 중복되는 두 SBS 중 모바일 사용자가 갈 확률이 높은 SBS를 파악하였다. 경로 확률을 통해 모바일 사용자가 머무를 가능성이 더 큰 SBS를 구별할 수 있는데, 해당 SBS의 인기도가 높은 파일 중에서 비교적 인기도가 낮은 파일들을 재배포할 용량만큼 제거하고 랜덤한 콘텐츠를 R_n 만큼 캐싱해준다. 인기도가 높은 파일이란 클러스터 H 에 속하는 파일을 의미한다. 클러스터 H 는 캐싱 알고리즘의 첫 번째 단계를 거친 이후 SBS에 캐싱된 파일을 인기도 순으로 정렬했을 때 연속적으로 저장된 파일이 형성하는 클러스터를 의미한다. 이 클러스터 H 에 속하면서 가장 낮은 인기도를 가진 파일부터 순차적으로 R_n 만큼씩 콘텐츠 크기를 제거하였다. 만약 클러스터에 내부에 있는 모든 파일에 대해 크기를 R_n 만큼씩 제거해도 총 재배포 해야 하는 용량에 도달하지 못했다면, 해당 과정을 반복해 재배포 용량만큼 모두 제거될 수 있을 때까지 계속 진행한다. 새로 캐싱할 파일은 다양한 콘텐츠가 포함될수록 MBS로부터 다운받는 데이터의 평균 크기가 줄어들기 때문에 랜덤하게 콘텐츠를 선택해 R_n 만큼 새로 저장하였다. 재배포 과정이 끝나면 인기 있는 콘텐츠의 파일의 일부 저장하면서도 다양한 파일을 함께 캐싱하기 때문에 모바일 사용자가 요구하는 파일이 없는 상황에서 서비스 영역이 중첩되는 다른 SBS를 사용해 더 유연하게 대처할 수 있게 된다.

4. 성능 평가

실험환경은 모바일 사용자가 이동하는 (6, 6)의 좌표평면에 4개의 SBS가 설치되었고 이 모든 SBS는 하나의 MBS의 영향을 받는다고 설정하였다. 각 SBS의 범위는 모두 동일하게 반경 100m를 가지고 있으며 좌표평면의 한 칸은 25m로, 모바일 사용자가 1.25m/s 속력으로 움직이면 20초가 소요

Table 2. Parameters used in experiments

Parameters	Values
Number of files	500
File size(B)	240MB
Transmission rate (R_n)	[4, 12]Mbps
SBS cache storage (C_n)	[10, 50]%
A number of requests	1000
Parameter of the zipf distribution	0.56

된다. 따라서 시간 단위(time slot)를 20초로 설정하였다. 이동 경로들 사이의 출발 지점과 도착하는 목표 지점은 동일하고 모든 경로는 같은 길이를 가지고 있어 이동에 걸리는 시간은 13 time slot로 모두 같게 설정했다. 갔던 길을 되돌아가지 않기 위해 위 또는 오른쪽으로만 이동할 수 있다고 가정하였고, 현재 위치에서 오른쪽으로 갈 확률을 0.7, 위로 갈 확률을 0.3으로 가정해 각 경로의 확률을 계산하였다. 기본적으로 콘텐츠의 인기도를 구하기 위해 파라미터가 0.56인 지프 분포(zipf distribution)를 사용하였다. 실험은 모바일 사용자가 출발 지점에서 특정 콘텐츠를 요구하고 목표 지점에 도착하기 위해 거치는 SBS들로부터 요구한 콘텐츠를 최대한 많이 제공받아 최종적으로 MBS에게 받아야 하는 데이터 양을 줄이는 것을 목표로 하였다. 실험에 사용한 파라미터들은 Table 2에 정리하였다.

제한한 알고리즘의 성능 평가를 위해 [5]에서 제시한 분산 탐욕 캐시 할당 알고리즘(Distributed Greedy Cache Allocation, DGCA)과 비교하였다. DGCA는 이동 경로의 확률과 파일의 인기도를 가중치 없이 계산에 사용해 캐싱을 진행하였고, 캐싱된 크기가 동일한 파일끼리 무리지어 R_n 만큼 늘이거나 줄여 파일을 재배포하였다. 이외에도 콘텐츠 파일의 인기도가 높은 순으로 캐싱하는 방법(popularity first, pop)을 비교에 사용하였다.

Fig. 1과 Fig. 2는 SBS의 캐싱 가능한 용량을 저장할 대상 전체 크기인 $B \times k$ 의 10%, 20%, 30%, 40%, 50%로 변경하면서 측정한 결과이다. Fig. 1을 보면 SBS의 캐싱 가능한 용량이 커지면서 모든 알고리즘이 전체적으로 MBS로부터 다운받는 평균 데이터 크기가 줄어든다는 것을 확인할 수 있었다. 제안 알고리즘은 인기도가 높은 파일과 인기도가 낮은 파일을 고루 저장하고 있는 반면, DGCA와 pop는 비교적 인기도가 높은 파일을 위주로 캐싱하기 때문에 제안한 알고리즘이 MBS로부터 제공받는 콘텐츠 파일의 크기가 더 작아졌다. Fig. 2는 모바일 사용자가 요구한 1000번의 콘텐츠 파일 요구에 SBS를 통해서만 모두 전달받는 비율을 나타낸 것이다. 비교 알고리즘들에 비해 제안 알고리즘은 파일이 작으므로 비교적 여러 SBS에 캐싱돼 있어 경로를 지나가면서 요구한 콘텐츠 파일의 크기 전체를 모두 다운받을 수 있는 확률이 높아졌음을 확인할 수 있었다.

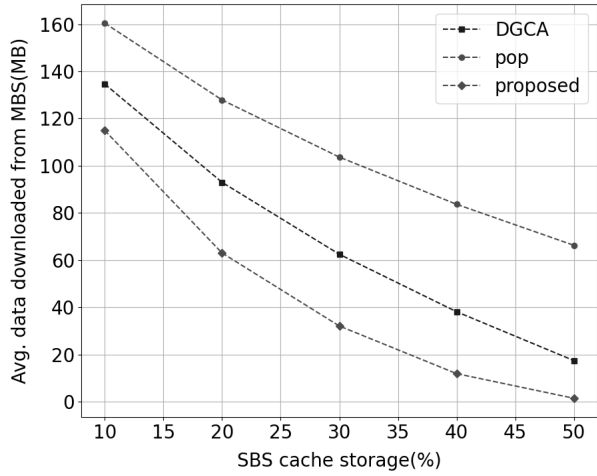


Fig. 1. Average Data Downloaded from MBS with Varying SBS Caching Storage

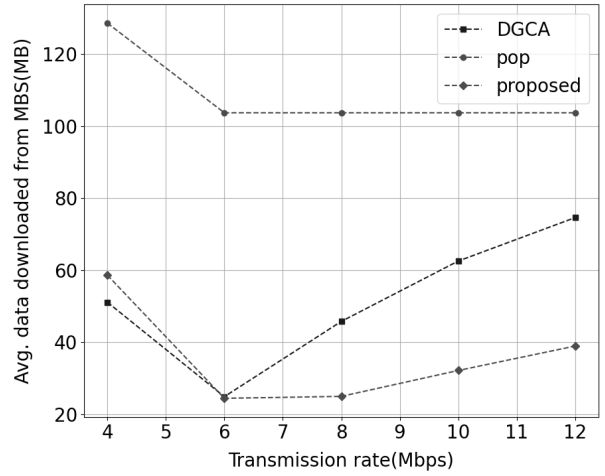


Fig. 3. Average Data Downloaded from MBS with Varying Transmission Rate

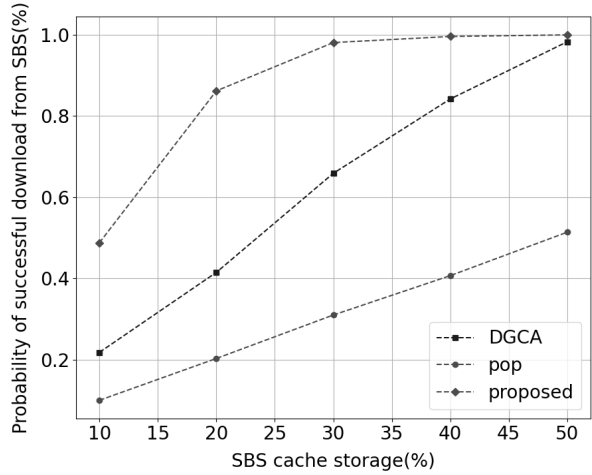


Fig. 2. Probability of Successful Download from SBS with Varying SBS Caching Storage

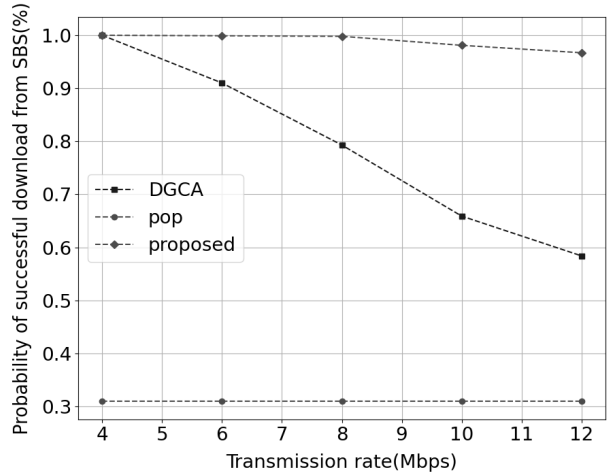


Fig. 4. Probability of Successful Download from SBS with Varying Transmission Rate

Fig. 3과 Fig. 4는 전송 속도를 4Mbps에서 12Mbps까지 2Mbps씩 증가시키며 측정한 결과이다. Fig. 3에서 전송 속도가 낮을 때는 DGCA와 큰 차이를 보이지 않는데 그 이유는 두 알고리즘 모두 여러 콘텐츠 파일에 대해 적은 양을 캐싱하기 때문이다. 하지만 전송 속도가 커지면 DGCA는 인기가 많은 파일에 캐싱이 몰리고 제안한 알고리즘은 상대적으로 고루 캐싱하여 MBS로부터 다운받는 데이터 양의 차이가 증가하였다. 6Mbps에서 감소하다 다시 증가하는 추세를 보이는데 그 이유는 전송 속도가 커지면서 T_{min} 과 T 의 차이가 커져 결과에 좋지 않은 영향을 미치기 때문이다. pop는 인기도가 높은 파일을 순차적으로 캐싱하여 비교적 전송 속도의 영향을 덜 받았다. Fig. 4는 모바일 사용자가 요구한 1000번의 콘텐츠 파일 요구에 대해 SBS를 통해서만 전달받는 비율을 나타내었다. DGCA와 제안한 알고리즘이 전송 속도가 커질수록 결과값이 줄어드는 이유는 큰 크기로 적은 종류의 콘텐츠 파일을 캐싱하기 때문이다. 제안한 알고리즘은 DGCA에

비해 다양한 콘텐츠 파일을 고루 가지고 있어 비교적 좋은 결과를 보였다. pop 방식은 전송 속도에 영향을 적게 받아 크게 변동 없는 모습을 보였다.

Fig. 5와 Fig. 6은 지프 분포의 파라미터를 0.2부터 0.8까지 0.2씩 증가시키면서 측정한 결과를 나타낸 것이다. 지프 분포의 파라미터가 1보다 작을 때 값이 커질수록 빈도가 쏠리는 현상이 높아지고, 값이 작을수록 고르게 분포한다. Fig. 5에서 모든 알고리즘이 지프 분포 파라미터가 커질수록 MBS로부터 다운받는 양이 감소하는 경향을 보였다. 이는 MBS로부터 다운받는 평균 데이터 양인 d_{av} 를 계산할 때 파일의 인기도 p_k 를 곱하는데 지프 분포 파라미터가 크면 적은 수의 파일에 인기도가 몰리고 나머지 많은 수의 콘텐츠 파일의 인기도가 매우 작아져 d_{av} 에 영향을 끼치는 정도가 줄어들기 때문이다. Fig. 6은 모바일 사용자가 요구한 1000번의 콘텐츠 파일 요구를 SBS를 통해 전송받은 비율을 나타냈다. 제안한 알고리즘과 pop 알고리즘은 지프 분포의 변화에도 성능이 일

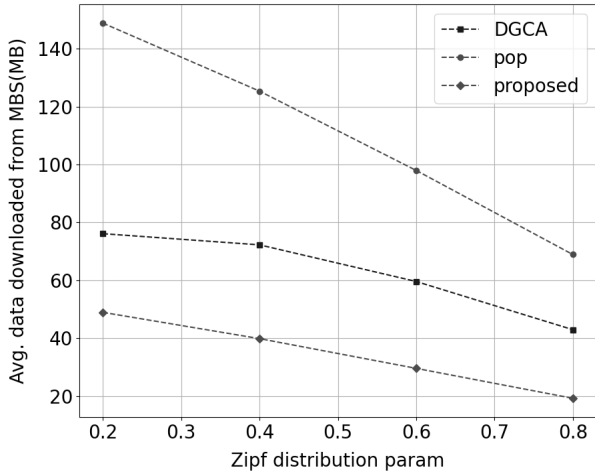


Fig. 5. Average Data Downloaded form MBS with Varying zipf Distribution Parameter

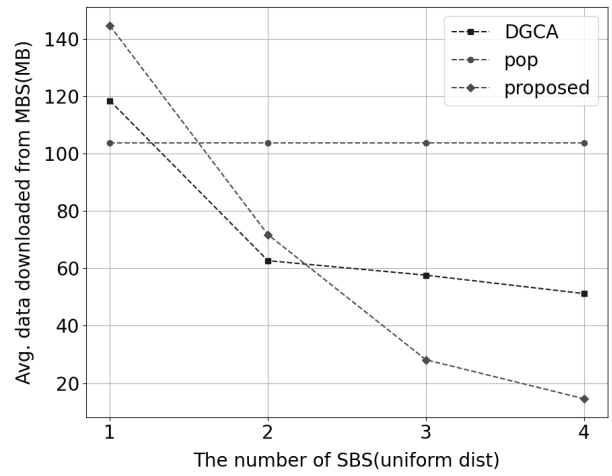


Fig. 7. Average Data Downloaded from MBS with Varying the Number of SBSs Arranged in Uniform Distribution

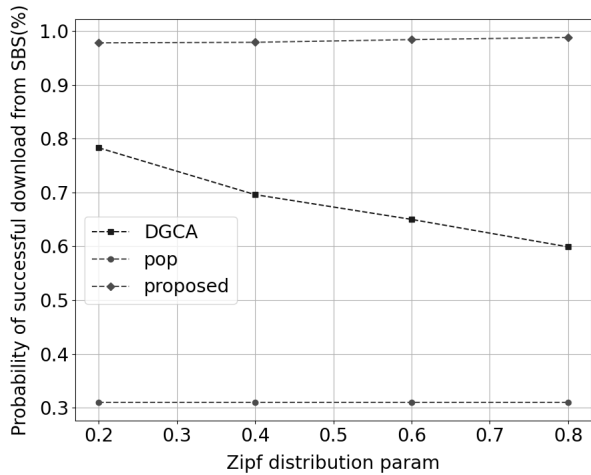


Fig. 6. Probability of Successful Download from SBS with Varying zipf Distribution Parameter

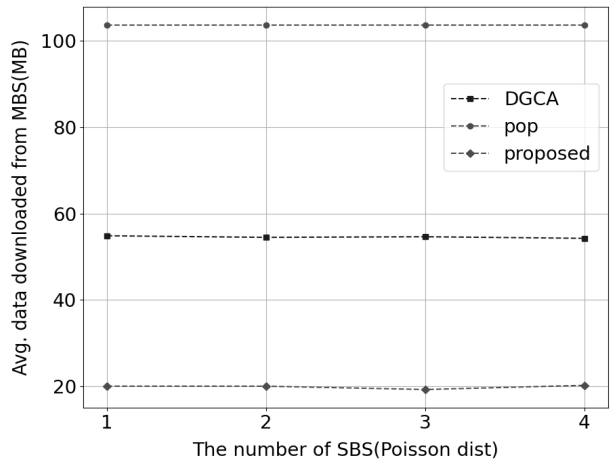


Fig. 8. Average Data Downloaded from MBS with Varying the Number of SBSs Arranged in Poisson Distribution

정한 반면 DGCA는 감소하는 모습을 보였다. 제안한 알고리즘은 여러 콘텐츠 파일을 고루 캐싱했기 때문에 모바일 사용자의 요구를 잘 수용할 수 있었고 pop 방식은 항상 인기도를 우선으로 캐싱하기 때문에 사용자의 요구를 제대로 반영하지 못해 낮은 결과를 보였다. DGCA는 인기도가 높은 특정 파일들에 대해서만 캐싱하는 비율이 높아져 사용자가 요구하는 콘텐츠 파일을 SBS를 통해 전달하는 확률이 낮아졌음을 확인할 수 있었다.

Fig. 7과 Fig. 8은 모바일 사용자가 접근할 수 있는 SBS 수가 평균적으로 일정한 상황에서 MBS로부터 전송받는 크기를 측정된 결과이다. Fig. 7은 모바일 사용자가 항상 동일한 수의 SBS에 접근할 때의 결과값으로 pop 방식은 MBS로부터 전송받는 양이 항상 일정한 반면 제안한 알고리즘과 DGCA는 모바일 사용자가 접근할 수 있는 SBS 수가 증가할수록 MBS로부터 다운받는 크기가 줄어드는 결과를 보였다. SBS의 수가 적을 때는 DGCA가, SBS의 수가 많을 때는 제안

한 알고리즘의 성능이 더 좋은 결과를 보였다. 그 이유는 제안한 알고리즘이 SBS의 수가 적을 때 캐싱된 파일의 종류가 다양하지만 조금씩 저장하고 있어 평균적으로 MBS로부터 다운받는 콘텐츠 파일의 크기가 증가하기 때문이다. 반대로 SBS 수가 많으면 사용자가 접근 가능한 SBS들에 캐싱된 파일이 다양하면서도 각 파일의 크기 총합이 커져 MBS로부터 다운받는 콘텐츠 파일 양이 줄어들었음을 확인할 수 있었다. Fig. 8은 모바일 사용자가 접근할 수 있는 SBS 수가 포아송 분포를 따를 때 MBS로부터 다운받는 데이터의 평균 크기를 나타내었다. SBS마다 캐싱한 콘텐츠 파일과 그 크기가 모두 다르기 때문에 모바일 사용자는 이동하면서 접근 가능한 SBS의 수가 많아질수록 많은 종류의 콘텐츠를 전송받을 확률이 높아지게 된다. SBS의 수가 포아송 분포를 따르게 되면 평균적으로 만나는 SBS 수는 동일하더라도 경로에 따라서 접근 가능한 SBS가 달라져 모바일 사용자는 여러 SBS에서 요구한 파일을 얻을 수 있었다. 따라서 MBS로부터 다운받는

콘텐츠 파일의 평균 크기를 줄이기 위해서는 다양한 SBS에 접근하는 것이 중요하다는 것을 확인할 수 있다.

5. 결 론

모바일 데이터 사용이 증가하여 발생하는 다양한 문제들을 해결하기 위해 클라우드보다 사용자와 가까이 위치한 엣지 서버 사용에 대한 관심이 높아졌다. 본 논문에서는 지연 오프로딩 스킴을 적용해 MBS로부터 전송받는 데이터 크기를 최소화하는 것을 목표로 이동하는 사용자를 위해 SBS에 콘텐츠 파일을 효율적으로 캐싱하는 기법을 제안하였다. 알고리즘의 첫 번째 단계에서 콘텐츠 파일의 인기도와 모바일 사용자의 이동 확률을 통해 각 SBS에 저장할 콘텐츠와 크기를 결정하였다. 두 번째 단계에서는 SBS의 서비스 영역이 중복되는 것을 고려해 영역이 많이 중첩될수록 더 많은 콘텐츠 파일을 재배치하였다. 제안한 알고리즘을 기존 DGCA 알고리즘과 비교한 결과 MBS로부터 다운받는 데이터 크기가 SBS 캐싱 용량에 따라 평균 약 50%, 전송 속도에 따라 평균 약 25% 감소한 것을 보였다. 또한 모바일 사용자가 요구한 콘텐츠를 SBS로부터 모두 전달받은 비율이 SBS 캐싱 용량과 전송 속도에 따라 각각 평균 1.6배 정도 높아진 모습을 확인하였다. 향후 연구에서는 모바일 사용자의 비디오 콘텐츠 요구를 예측하는 모델을 도입하여 불확실한 미래의 요구에 유연하게 대처할 수 있는 캐싱 기법을 다루고자 한다.

References

- [1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022," [Internet], <http://media.mediapost.com/uploads/CiscoForecast.pdf>.
- [2] R. Yang and S. Guo, "A mobile edge caching strategy for video grouping in vehicular networks," *2021 13th International Conference on Advanced Computational Intelligence (ICACI)*, Wanzhou, China, 2021.
- [3] Z. Yu, J. Hu, G. Min, H. Xu, and J. Mills, "Proactive content caching for internet-of-vehicles based on peer-to-peer federated learning," *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, Hong Kong, 2021.
- [4] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," *IEEE Journal on Selected Areas in Communications*, Vol.35, No.5, pp.1076-1089, 2017.
- [5] E. Ozfatura and D. Gündüz, "Mobility and popularity-aware coded small-cell caching," *IEEE Communications Letters*, Vol.22, No.2, pp.288-291, 2018.
- [6] S. Park, S. Oh, Y. Nam, J. Bang, and E. Lee, "Mobility-aware distributed proactive caching in content-centric vehicular networks," *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, Paris, France, 2019.
- [7] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Transactions on Intelligent Transportation Systems*, Vol.22, No.8, pp.5341-5351, 2021.
- [8] L. Yao, A. Chen, J. Deng, J. Wang, and G. Wu, "A cooperative caching scheme based on mobility prediction in vehicular content centric networks," *IEEE Transactions on Vehicular Technology*, Vol.67, No.6, pp.5435-5444, 2018.
- [9] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Co-operative content caching in 5G networks with mobile edge computing," *IEEE Wireless Communications*, Vol.25, No.3, pp.80-87, 2018.
- [10] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Communications Surveys & Tutorials*, Vol.21, No.3, pp.2525-2553, 2019.
- [11] A. Mahmoo, C. E. Casetti, C. F. Chiasserini, P. Giaccone, and J. Härri, "The RICH prefetching in edge caches for in-order delivery to connected cars," *IEEE Transactions on Vehicular Technology*, Vol.68, No.1, pp.4-18, 2019.
- [12] S. Park, Y. Shin, Y. Nam, J. Bang, and E. Lee, "Proactive caching protocol based on trajectory transition probability in vehicular networks," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp.581-582, 2019.
- [13] S. Zhang, W. Sun, and J. Liu, "Spatially cooperative caching and optimization for heterogeneous network," *IEEE Transactions on Vehicular Technology*, Vol.68, No.11, pp.11260-11270, 2019.
- [14] H. Im and C. Hong, "A research on machine learning based mobility prediction method for content caching," *Proceedings of Symposium of the Korean Institute of Information Scientists and Engineers*, pp.1097-1099, 2018.
- [15] A.-T. Tran, T.-V. Nguyen, V.-D. Tuong, N.-N. Dao, and S. Cho, "On stalling minimization of adaptive bitrate video services in edge caching systems," *2020 International Conference on Information Networking (ICOIN)*, Barcelona, Spain, 2020.
- [16] K. Poularakis and L. Tassiulas, "Code, cache and deliver on the move: A novel caching paradigm in hyper-dense small-cell networks," *IEEE Transactions on Mobile Computing*, Vol.16, No.3, pp.675-687, 2017.



최 윤 정

<https://orcid.org/0000-0003-1169-1102>
e-mail : potatochips@sookmyung.ac.kr
2021년 숙명여자대학교 소프트웨어융합과
(학사)
2021년 ~ 현 재 숙명여자대학교 IT공학과
석사과정

관심분야: 캐싱, VANET, 강화학습



임 유 진

<https://orcid.org/0000-0002-3076-8040>
e-mail : yujin91@sookmyung.ac.kr
2013년 일본 Tohoku University.
Department of Information
Sciences(박사)
2000년 숙명여자대학교 전산학과(박사)

2004년 ~ 2015년 수원대학교 정보미디어학과 부교수
2016년 ~ 현 재 숙명여자대학교 IT공학과 교수
관심분야: 지능형 시스템, IoT, Edge Computing