

# 차량 환경에서 엣지 커버리지 오버랩을 고려한 강화학습 기반의 엣지 캐싱

최윤정, 임유진  
숙명여자대학교 IT공학과  
[potatochips@sookmyung.ac.kr](mailto:potatochips@sookmyung.ac.kr), [yujin91@sookmyung.ac.kr](mailto:yujin91@sookmyung.ac.kr)

## Edge Caching Based on Reinforcement Learning Considering Edge Coverage Overlap in Vehicle Environment

Yoonjeong Choi, Yujin Lim,  
Dept. of IT Engineering, Sookmyung Women's University

### 요 약

인터넷을 통해 주위 사물과 연결된 차량은 사용자에게 편리성을 제공하기 위해 다양한 콘텐츠를 요구하는데 클라우드로부터 가져오는 시간이 비교적 오래 걸리기 때문에 차량과 물리적으로 가까운 위치에 캐싱하는 기법들이 등장하고 있다. 본 논문에서는 기반 시설이 밀집하게 설치된 도시 환경에서 maximum distance separable(MDS) 코딩을 사용해 road side unit(RSU)에 캐싱하는 방법에 대해 연구하였다. RSU의 중복된 서비스 커버리지 지역을 고려하여 차량의 콘텐츠 요구에 대한 RSU hit ratio를 높이기 위해 deep Q-learning(DQN)를 사용하였다. 실험 결과 비교 알고리즘보다 hit ratio 측면에서 더 높은 성능을 보이는 것을 증명하였다.

### 1. 서론

전 세계적으로 인터넷 사용이 증가하면서 동시에 사물인터넷(IoT, Internet of Things)이 많은 분야에 적용되고 있다. 사물인터넷이란 상호 운용 가능한 정보와 통신 기술을 바탕으로 물리적 그리고 가상의 사물들이 서로 연결된 세계적인 인프라를 의미한다 [1]. 사물인터넷이 적용된 분야 중 차량은 2018년부터 2023년까지 연평균 성장률을 30%로 예측하였는데 집, 도시, 건강 등 다른 카테고리에 비해 높은 수치를 기록했다[2]. 사물인터넷이 적용된 차량은 사용자에게 편리성을 제공하는데 예를 들어 차량 내부에서 다양한 콘텐츠 감상, 최적의 길 안내 등이 있다.

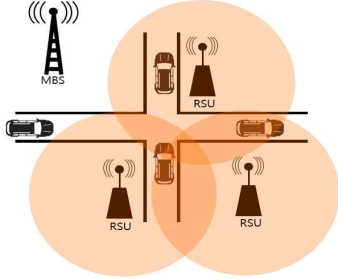
하지만 이동하는 차량 내부에서 동영상이나 게임과 같은 콘텐츠를 사용할 때 크기가 큰 콘텐츠를 클라우드로부터 전송받기에는 시간이 오래 걸린다는 문제점이 있다. 해결 방법으로 모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC)이 고려되고 있다. MEC는 차량과 지리적으로 가까이 위치하여 차량에게 컴퓨팅과 저장 공간을 제공할 수 있다. 특히 MEC에 있는 저장 공간을 활용해 콘텐츠 캐싱 노드(caching node)로 사용하면 전송 데이터 트래픽을

완화시키고 콘텐츠 전송 지연을 줄일 수 있다[3]. 캐싱 노드로 사용할 수 있는 대표적인 시설로는 small base station(SBS), macro base station(MBS), RSU 등이 있다. 하지만 MEC는 클라우드보다 작은 저장 용량을 가지고 있어 어떤 콘텐츠를 얼마나 캐싱할지 결정하는 것이 매우 중요하다. 또한 차량은 빠른 속도로 계속 움직이므로 차량의 이동성을 고려해야 한다. [4]는 움직이는 차량을 클러스터링하고 long short-term memory(LSTM)를 통해 차량이 요구한 콘텐츠를 예측하는 모델을 만들어 Q 러닝(Q-learning)을 통해 hit ratio를 높이는 최적의 RSU 캐싱 기법을 연구하였다. [5]는 차량을 위해 RSU에 캐싱과 컴퓨팅 자원을 분배하는데 시스템 비용을 최소화하는 것을 목표로 하여 차량의 이동성과 서비스 지연 제한 시간을 고려해 DQN을 사용하였다. [6]은 MDS 코딩을 캐싱에 적용해 MBS가 아닌 SBS로부터 콘텐츠를 최대한 전달 받도록 Q 러닝 기반 알고리즘을 연구하였다.

도시 환경은 차량 이동량과 기반 시설이 많아 간섭 현상이 빈번하게 일어나고 RSU의 범위가 서로 겹치기도 한다. 따라서 도시 환경에서 차량을 위해

RSU에 캐싱을 할 때, 중복된 RSU 범위를 고려해야 한다. 본 논문에서는 RSU의 범위가 중복된 환경에서 hit ratio를 높일 수 있는 RSU 캐싱 기법에 대해 연구하고자 한다.

## 2. 제안 기법



( 그림 1) RSU가 밀집된 환경

### 2.1 시스템 모델

하나의 MBS와  $M$ 개의 RSU로 이루어진 이중 네트워크를 가정한다. MBS와 RSU들 사이는 유선으로 연결돼 있으며 MBS 또는 RSU와 차량은 무선으로 통신한다. MBS에는 콘텐츠  $K$ 개가 모두 존재한다고 가정하였다. 이동하는 차량은 RSU에 캐싱된 콘텐츠들을 제공받으며 해당 콘텐츠가 없을 경우 MBS부터 전송받는다. 밀집하게 모인 RSU 서비스 커버리지가 중복되는 공간의 간섭을 고려하기 위해 signal-to-interference-plus-noise ratio(SINR)을 사용하였다.

차량의 이동 속도가 매우 빠르기 때문에 차량의 이동 속도와 캐싱 업데이트 속도를 다르게 적용하였다. 차량의 이동과 콘텐츠 다운로드드는 작은 타임 스케일(small time scale)을 사용하고, 캐싱 업데이트와 콘텐츠 요구는 큰 타임 스케일(large time scale)을 사용했다. 차량의 이동 속도가 빨라 차량이 RSU 서비스 지역 머물 경우 다운받을 수 있는 콘텐츠 크기가 한정적이기 때문에 콘텐츠의 전체 크기를 캐싱하는 것보다 콘텐츠를 작은 조각으로 나눠서 캐싱하는 것이 더 효율적이다. 특히, rateless MDS 코딩을 사용하면 콘텐츠 조각 순서에 상관 없이 파일의 parity symbol의 수를 고려하면 콘텐츠 파일을 복구할 수 있기 때문에[7] rateless MDS 코딩을 사용한다고 가정하였다. 차량은 시간에 따라 위치하고 있는 RSU로부터 SINR을 통해 전송 속도를 구해 해당 콘텐츠를 전송받는다.

### 2.2 문제 정의

본 논문의 목표는 이동하는 차량이 콘텐츠를 요구

할 때 제한 시간 내에 콘텐츠를 전달하면서 hit ratio를 최대화 하는 것이다.

큰 타임 스케일의 집합을  $T$ , 차량의 집합을  $J$ 라고 할 때, 시간  $t$ 에 차량  $j$ 가 요구한 횟수를  $s_j^t$ 라 하고 RSU로부터 콘텐츠를 전달받은 여부를  $h_j^t \in \{0, 1\}$ 라 하면 시간  $t$ 에서의 hit ratio  $H^t$ 는 다음과 같이 표현 가능하다.

$$H^t = \sum_{j=1}^J \frac{h_j^t}{s_j^t} \quad (1)$$

RSU  $m$ 에 저장할 수 있는 최대 용량을  $c_m$ 이라고 할 때, hit ratio를 최대화 하는 문제는 다음과 같이 정의될 수 있다.

$$P1: \max \sum_{t=1}^T H^t$$

subject to:

$$x_{m,k}^t \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}, \quad (2)$$

$$\sum_{k=0}^K x_{m,k}^t \leq c_m, \forall m. \quad (3)$$

$x_{m,k}^t$ 은 RSU  $m$ 에 캐싱된 콘텐츠와 캐싱된 정도를 표시한다. 예를 들어,  $x_{m,k}^t = 0.2$ 이라면 RSU  $m$ 에 콘텐츠  $k$ 가  $k$ 의 전체 크기 중 20% 만큼 캐싱되어 있다는 의미이다.

### 2.3 알고리즘

차량이 계속 이동하는 환경에서는 이동성을 정확하게 예측하기 어렵고 차량이 빠르게 움직이기 때문에 최적의 캐싱 기법을 계산하게 되면 계산 복잡도가 높아진다. 따라서 본 논문에서는 동적으로 변화하는 환경을 반영하기 위하여 DQN을 사용하였다. DQN은 Q 러닝의 발전된 형태로, 에이전트가 주변 환경을 살펴면서 행동을 취하고 그로 인해 바뀐 상태를 관찰하며 보상을 크게 가지는 방향으로 학습하는 강화학습의 한 종류이다. Q 러닝과 다르게 뉴럴 네트워크를 사용해 보상을 계산하고 업데이트한다.

에이전트는 MBS에 위치해 있다고 가정하며, 환경 내 RSU들로부터 정기적으로 상태메시지를 수신하여 현재 상태를 인식할 수 있다고 가정하였다.

#### 2.3.1 상태 공간

시간  $t$ 에서 상태  $s_t$ 는 다음과 같이 정의하였다.

$$s_t = \{g_{n,k}^t, v_n^t, q_{m,k}^t\} \quad (4)$$

$g_{n,k}^t$ : 서비스 커버리지가 중복되는 RSU들을 클래스

터링한 집합을  $N$ 이라고 할 때, 시간  $t$ 에서 클러스터  $n$  내부에 있는 RSU에 캐싱된 콘텐츠 크기 합을 의미한다.

$$g_{n,k}^t = \sum_{m \in n} x_{m,k}^t \quad (5)$$

$g_{n,k}^t$ 는 클러스터  $n$ 에 콘텐츠  $k$ 가 얼마나 저장되어 있는지를 말하며, 만약 같은 콘텐츠가 동일 클러스터에서  $k$ 의 크기 이상 캐싱 되었을 경우 초과된 크기만큼 제거하고 다른 콘텐츠를 캐싱할 때 hit ratio가 더 좋아질 수 있다. RSU가 밀집한 환경일수록 다양한 콘텐츠를 가지고 있어야 높은 hit ratio를 만들어낼 수 있기 때문이다[8].

$v_n^t$ : 시간  $t$ 에서 클러스터  $n$  안에 존재하는 차량 수를 의미한다.  $t$  시점에 차량은 콘텐츠 요구를 하기 때문에 클러스터  $n$ 에서 발생된 총 콘텐츠 요구 수는 차량 수와 비례한다. 차량의 위치에 따라 여러 RSU에 접근할 수 있어서 단일 RSU가 아닌 클러스터 단위로 차량 수를 계산하였다.

$q_{m,k}^t$ : 시간  $t$  동안 RSU  $m$ 에게 요구하는 콘텐츠  $k$ 의 횟수를 의미한다.  $t$ 이전 시간에 대한 내용을 누락하기 때문에 차량이 요구하는 콘텐츠를 확인해 RSU  $m$ 에서 콘텐츠 인기도를 파악할 수 있다.

### 2.2.2 행동

에이전트가 시간  $t$ 에서 할 수 있는 행동은 다음과 같다.

$$a_t = \{x_{m,k}^t\} \quad (6)$$

$x_{m,k}^t$ : 시간  $t$ 일 때, RSU  $m$ 에 콘텐츠  $k$ 의 캐싱 비율을 결정한다. MDS 코딩을 가정하기 때문에 콘텐츠들은 작게 나눈 상태로 캐싱할 수 있어  $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$  중 하나의 값을 콘텐츠의 캐싱할 비율로 가진다. RSU  $m$ 에 캐싱된 콘텐츠들은 RSU의 용량 이상으로 캐싱할 수 없으며 앞서 언급한 상태들을 관찰하면서 보상을 최대화 하는 방향으로 행동을 정하였다. 다만 항상 최대 보상을 쫓으면 국소 최적해(local optimal solution)에 머무를 가능성이 있어 일정 확률로 랜덤한 행동을 선택하였다.

### 2.2.3 보상

에이전트가 행동을 통해 이루고자 하는 목표는 hit ratio를 최대화 하는 것이므로 보상은 다음과 같다.

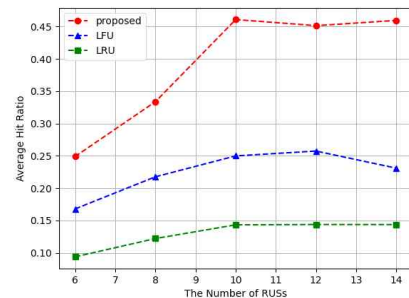
$$H^t = \sum_{j=1}^J \frac{h_j^t}{s_j^t} \quad (1)$$

### 3. 성능평가

성능 평가를 위해 실제 데이터를 가지고 실험을 진행하였다. ECML PKDD 2015 Predict Challenge에서 제공된 포르투갈 택시 서비스 궤적 데이터를 사용해 실제 차량 움직임을 고려하였다. 위도와 경도로 이루어진 차량 좌표를 이용해 가로 1km, 세로 1km로 실험 공간을 정의하고, 해당 지역 내부를 돌아다니는 차량만을 정제해 사용하였다. 실험에서 가정한 총 콘텐츠 수는 100개, RSU 용량은 전체 콘텐츠 크기의 10%만 캐싱할 수 있도록 가정하였다. 차량이 콘텐츠를 요구하는 모델은 지프 분산(zipf distribution)을 사용했다. 지프 분산의 파라미터 값이 0과 1사이일 때, 0에 가까울수록 모든 콘텐츠를 비슷한 빈도로, 1에 가까울수록 인기가 높은 콘텐츠를 우선으로 요구하게 된다. 성능 평가를 위해 Least Recently Used(LRU), Least Frequently Used(LFU)와 비교하였다. LRU는 가장 최근에 요구된 콘텐츠를 위주로 캐싱하고 LFU는 가장 자주 요구되는 콘텐츠를 위주로 캐싱하는 기법이다. 실험에서 사용한 파라미터는 <표 1>에 정리하였다.

<표1> 실험에서 사용된 파라미터 값

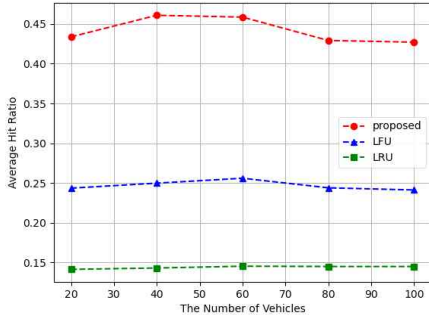
Parameter	Value	Parameter	Value
K(the number of contents)	100	M(the number of RSUs)	[6, 14]
large time slot(minutes)	1	small time slot(seconds)	15
episodes	500	zipf parameter	0.56



( 그림 2 )RSU 수에 따른 평균 hit ratio

그림 2는 RSU 개수를 늘려 가면서 측정한 평균 hit ratio이다. 실험 공간의 범위를 제한했기 때문에 RSU의 수가 많아질수록 사용자가 RSU 서비스 커버리지가 오버랩되는 부분에 위치해있을 확률이 높아지고 다수의 RSU로 접근이 가능하여 hit ratio가 증가하게 된다. 제안한 알고리즘의 경우 다른 알고

리즘에 비해 서비스 커버리지가 겹친 인접한 RSU 사이의 콘텐츠 중복 정도를 줄였기 때문에 그만큼 다양한 콘텐츠를 저장할 수 있게되어 더 좋은 성능을 보였다. RSU 수가 10개 이상일 경우 서비스 커버리지가 중복되는 지역이 많아졌지만 중복을 고려해야 하는 RSU도 함께 증가하기 때문에 hit ratio가 더 이상 늘지 않는 모습을 보였다.



( 그림 3 ) 차량 수에 따른 평균 hit ratio

그림 3은 실험 환경에 존재하는 차량의 수를 증가시키면서 측정한 평균 hit ratio이다. 제안한 알고리즘은 차량이 20대일 때 비교적 낮은 hit ratio를 보였는데 이는 전체 콘텐츠 요구량이 많지 않아 개별 콘텐츠 요구 hit/miss에 크게 영향을 받기 때문이다. 반대로 차량이 80-100대일 때 비교적 낮은 값을 갖는 이유는 콘텐츠 요구가 많아지면 다양한 콘텐츠를 요구할 가능성이 높아졌지만 RSU의 캐싱 용량은 한정되어 있기 때문이다.

#### 4. 결론

본 논문에서는 RSU가 밀집된 지역에서 차량의 이동성을 고려해 차량이 요구하는 콘텐츠를 RSU로부터 전송받는 hit ratio를 높이는 기법을 제안하였다. RSU의 서비스 커버리지가 겹치는 지역을 클러스터링해, 클러스터 내에 RSU들에 저장된 콘텐츠의 중복성을 낮추고 이를 통하여 보다 다양한 콘텐츠들이 클러스터 내에 저장되게 하였다. 이를 위하여 각 클러스터링에 존재하는 차량 수와 콘텐츠 카운트, 각 RSU에게 요구한 각 콘텐츠 횟수를 상태로 고려하고 보상으로 hit ratio를 설정해 각 RSU에 무슨 콘텐츠를 얼마나 캐싱할지 결정하는 DQN기반 알고리즘을 설계하였다. 실험 결과 비교 알고리즘에 비해 RSU 수에 따라 LFU와 평균 72% 정도, LRU와 평균 190% 정도 높은 성능을 보였다. 차량 수에 따라 RFU와 평균 78% 정도, LRU와 200% 정도 높은 성능을 보였다.

#### 사사문구

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2021R1F1A1047113).

#### 참고문헌

- [1] 표철식, 강호용, 김내수, 방효찬, "IoT(M2M) 기술 동향 및 발전 전망," Information & communications magazine v.30 no.8 , 2013, pp. 3-10
- [2] "Cisco Annual Internet Report(2018 - 2023)," Cisco, 2020
- [3] J. Yao, T. Han and N. Ansari, "On Mobile Edge Caching," in IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2525-2553, thirdquarter 2019
- [4] R. Wang, Z. Kan, Y. Cui, D. Wu and Y. Zhen, "Cooperative Caching Strategy With Content Request Prediction in Internet of Vehicles," in IEEE Internet of Things Journal, vol. 8, no. 11, pp. 8964-8975, Jun. 2021
- [5] L. T. Tan and R. Q. Hu, "Mobility-Aware Edge Caching and Computing in Vehicle Networks: A Deep Reinforcement Learning," in IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 10190-10203, Nov. 2018
- [6] S. Gao, P. Dong, Z. Pan and G. Y. Li, "Reinforcement Learning Based Cooperative Coded Caching Under Dynamic Popularities in Ultra-Dense Networks," in IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5442-5456, May 2020
- [7] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch and G. Caire, "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers," in IEEE Transactions on Information Theory, vol. 59, no. 12, pp. 8402-8413, Dec. 2013
- [8] P. Lin, Q. Song and A. Jamalipour, "Multidimensional Cooperative Caching in CoMP-Integrated Ultra-Dense Cellular Networks," in IEEE Transactions on Wireless Communications, vol. 19, no. 3, pp. 1977-1989, March 2020