

# 주차 할당 만족도와 네트워크 오버헤드를 고려한 모바일 엣지 컴퓨팅 기반 스마트 주차 시스템

이 경 민\*, 이 미 정°

## MEC-Based Smart Parking System Considering User Satisfaction and Network Overhead

Kyoung-min Lee\*, Mee-jeong Lee°

### 요 약

스마트 주차 시스템은 신속하게 운전자의 요구사항을 최대한 만족하는 주차장을 할당하는 것을 목표로 한다. 일반적으로 주차장은 차량의 목적지 근처로 국한된 지역에서 선택된다는 특성에 착안하여, 본 논문에서는 중앙집중 서버 없이 지역적으로 분포된 Mobile Edge Computing (MEC) 서버들 간의 협력을 기반으로 하는 주차장 할당 방안을 제안하였다. 또한 주차장 할당 과정에서 운전자 요구사항에 대해 각 주차장이 만족하는 정도를 점수로 나타내는 방법을 제안하였다. 네트워크 시뮬레이터인 NS-3를 사용하여, 제안하는 방안과 기존 방안을 비교한 결과, 제안하는 방안이 높은 주차장 할당 만족도를 유지하면서, 작은 오버헤드와 짧은 지연을 갖는 것을 보였다.

**키워드** : 클라우드, 모바일 엣지 컴퓨팅, 스마트 주차 시스템

**Key Words** : Cloud, Mobile Edge Computing, Smart Parking System

### ABSTRACT

The smart parking systems aim to quickly allocate parking lot that meets driver's requirements as much as possible. This paper proposed a parking lot allocation system based on cooperation between locally distributed Mobile Edge Computing (MEC) servers without a centralized server, leveraging the characteristics that parking lots are selected in areas geographically limited to the vehicle's destination. In addition, we proposed a method of expressing the degree to which each parking lot satisfies the driver's requirements in the parking lot allocation process as a score. Using NS-3, a network simulator, we measured the level of user satisfaction with the parking lot allocation, and the network overhead. Comparing the proposed system with the existing ones, it is shown that the proposed system has smaller overhead and shorter delay while maintaining higher level of user satisfaction with the parking lot allocation.

### I. 서 론

차량의 수가 꾸준히 증가하면서, 운전자들이 빈 주차 공간을 찾기가 더욱 어려워지고 있다. 운전자들이 목적

지에 도착한 후, 바로 주차하지 못하고 주차 공간을 찾기 위한 추가적인 주행을 하게 되면 탄소배출량 증가로 이어져 환경오염에 영향을 끼치고, 이를 처리하기 위한 사회적 비용을 발생시킨다.<sup>[1]</sup> 또한 주차 공간을 찾는

\* First Author : Ewha Womans University, Department of Computer Science and Engineering, jully0425@ewhain.net, 학생회원

° Corresponding Author : Ewha Womans University, Department of Computer Science and Engineering, lmj@ewha.ac.kr, 종신회원  
 논문번호 : 202402-029-D-RN, Received February 7, 2024; Revised April 11, 2024; Accepted May 11, 2024

차량들이 증가할수록 교통은 더욱 혼잡해진다. 이러한 문제를 해결하기 위해 스마트 주차 시스템이 제안되었다. 스마트 주차 시스템은 차량이 목적지에 도착하기 전, 시스템에서 저장하고 있는 주차장의 정보와 차량으로부터 받은 차량의 요청 정보를 바탕으로 차량에 대해 주차장을 할당한다. 이러한 스마트 주차 시스템에 대해 최근 다양한 구조의 시스템이 제시되었다.

클라우드가 주차장 할당을 진행하는 구조로, MAPark<sup>[2]</sup>, ASPIRE<sup>[3]</sup>, 오픈 플랫폼 기반의 스마트 주차 시스템<sup>[4]</sup>이 있다. 이들 시스템에서는 공통적으로 클라우드에 서비스 지역 내의 모든 주차장의 실시간 정보가 저장되고, 차량의 주차장 할당 요청을 받은 클라우드 서버는 이들 정보를 모두 활용하여 서비스 구역 내의 주차장 가운데 할당 목적에 따른 최적의 주차장을 할당한다. 따라서 클라우드 기반의 주차 시스템은 운전자가 가장 만족할 수 있는 주차장을 할당할 수 있다는 장점을 가진다. 하지만 클라우드까지 주차장 정보가 업데이트 되어야 하고, 주차장 할당 요청이 전송되어야 하기 때문에, 오버헤드가 크고, 지연이 길다는 단점이 있다.

O. Tran Thi Kim et al.<sup>[5]</sup>은 차량의 목적지와 가장 가까운 한 개의 Road Side Unit (RSU)이 주차장 할당을 진행하는 구조를 제안하였다. 각 RSU는 담당 구역 내에 있는 모든 주차장의 정보를 수집하여 저장하고, 차량은 현재 위치에서 가장 가까운 RSU로 주차장 할당 요청을 전송한다. 만약 차량이 목적지와 충분히 가까운 곳에서 주차장 할당을 전송하여, 차량의 현재 RSU와 목적지를 담당하는 RSU가 동일하다면, 해당 RSU에서 주차장 할당이 진행되어 빠르게 주차장을 할당할 수 있다. 하지만 두 RSU가 동일하지 않다면, 차량의 주차장 할당 요청은 차량의 현재 RSU에서 모든 RSU의 위치 정보를 저장하고 있는 클라우드를 거쳐서 목적지를 담당하는 RSU로 전송된 후 주차장 할당이 진행된다. 따라서 이 경우에는 오버헤드가 크고, 지연이 길다는 단점이 있다. 그리고, 차량의 목적지를 담당하는 RSU가 담당하는 구역이 차량의 목적지를 중심으로 운전자가 설정한 최대 허용 거리를 반경으로 하는 주차장 검색 구역을 모두 포함하지 않을 수 있으므로, 운전자를 만족시킬 수 있는 최적의 주차장을 찾지 못할 수 있다.

C. Tang, X et al.<sup>[6]</sup>에서는 RSU로부터 받은 주차장 정보를 활용해 차량이 직접 주차장을 선택해서 주차장 할당을 요청하는 구조를 제안하였다. 각 주차장에 설치된 Fog node는 주차장 정보를 수집해서 저장하고, 클라우드와 가장 가까운 RSU에게 주기적인 비콘 메시지로 주차장 정보를 전달한다. 클라우드는 이 정보를 저장하고, RSU는 비콘 메시지에서 수집한 주차장 정보를

주변 차량에게 브로드캐스트한다. 차량은 비콘으로부터 수집한 주차장 가운데 원하는 주차장을 선택하여, RSU를 통해 해당 주차장의 Fog node로 주차장 할당 요청을 전송한다. 차량이 도착하기 전까지 해당 주차장에 빈 공간이 있다면, Fog node가 빠르게 주차장을 할당할 수 있다. 빈 공간이 없다면 Fog node는 클라우드로 주차장 할당 요청을 전송하고, 이를 받은 클라우드가 서비스 영역 내의 모든 주차장을 고려하여 새로운 주차장을 할당함으로써, 차량이 주차장을 할당 받지 못하는 경우를 방지한다. 하지만 주차장 정보를 제공하기 위한 주기적인 비콘 메시지의 잦은 전송으로 인하여 네트워크 트래픽 오버헤드가 크다는 문제가 있다. 그리고 이 구조 또한 [5]와 유사하게 차량이 한 개의 RSU로부터 주차장 정보를 받기 때문에, 운전자가 가장 만족할 수 있는 주차장 정보는 제외될 수 있다는 한계를 가진다.

이에, 본 논문에서는 선행 연구를 기반으로, 클라우드를 사용하지 않으면서, 운전자가 최대로 만족할 수 있는 주차장을 할당하는 것을 목표로, MEC 기반의 스마트 주차 시스템을 제안한다.<sup>[7]</sup> 또한 선행 연구에서 나아가 주차장 할당 과정에서 주고받는 패킷을 제안하고, 이에 포함된 정보를 활용하여 목적지 주변의 MEC들의 협력을 통해 주차장을 할당하는 알고리즘을 제안한다. 주차장 할당 시, 주차장 내 주차 공간 할당은 고려하지 않는다. 그리고 네트워크 시뮬레이터인 NS-3를 사용하여, 제안하는 방안에서의 주차장 할당 성능, 오버헤드, 지연을 분석하고 기존 방안들과 비교한다.

2장에서는 제안 방안과의 성능을 비교할 관련 연구의 동작 방식에 대해서 좀 더 자세히 설명하고, 3장에서는 제안하는 방안의 구성 요소 및 동작 방식에 대해 설명한다. 4장에서는 실험을 통한 결과를 분석하고, 5장에서 결론을 맺는다.

## II. 관련 연구

2장에서는 본 논문의 4장 실험에서 제안하는 방안과의 성능을 비교할 3가지 관련 연구의 동작 방식에 대해서 설명한다.

MAPark<sup>[2]</sup>에서는 클라우드에서 주차장 할당이 진행되며, 시스템은 그림 1과 같이 차량, 클라우드, 주차장으로 구성된다. 클라우드는 차량과 데이터를 주고받는 요청 센터, 각 차량에게 주차장을 할당하는 할당 센터, 모든 주차장의 정보를 저장하는 자원 관리 센터로 구성된다. 각 주차장은 차량의 출입으로 인해 주차장 정보에 변화가 있을 때마다 클라우드로 주차장 정보를 전송하여, 클라우드에 저장된 주차장 정보를 갱신한다. 차량은

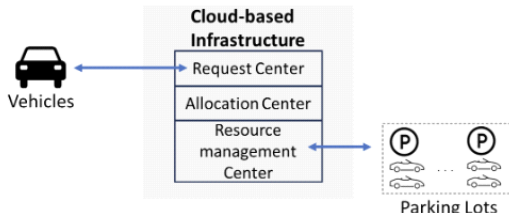


그림 1. MAPark<sup>[2]</sup>의 구성 요소  
Fig. 1. Components of MAPark<sup>[2]</sup>

가장 가까운 기지국을 통해 클라우드로 주차장 할당 요청을 전송한다. 이를 받은 클라우드는 저장하고 있는 서비스 영역 내의 모든 주차장 정보를 활용하여 차량에 대해 주차장을 할당하고, 그 결과를 차량에게 전송한다. 운전자가 할당된 주차장을 최종적으로 선택하면, 차량은 클라우드로 이를 알려서 할당을 완료한다.

O. Tran Thi Kim et al.<sup>[5]</sup>에서는 차량의 목적지와 가장 가까운 한 개의 RSU에서 주차장 할당이 진행되며, 시스템은 그림 2와 같이 Fog node, RSU, 클라우드, 차량으로 구성된다. 각 주차장에 설치된 Fog node는 주차장 정보를 수집하고 변경 사항이 있는 경우, 주차장이 위치한 구역을 담당하는 RSU로 주차장 정보를 전송한다. 따라서, 각 RSU는 각각의 담당 구역 내에 있는 주차장의 정보를 저장한다. 그리고, 클라우드는 모든 RSU의 위치를 저장하고 있다. 각 차량은 현재 위치에서 가장 가까운 RSU에게 주차장 할당 요청을 전송한다. 차량의 현재 위치를 담당하는 RSU가 차량의 목적지 위치를 담당하는 RSU와 일치하는 경우, 해당 RSU가 담당 구역 내 주차장 중에서 할당을 진행하고, 결과를 차량에게 전송한다. 이는 그림 2에서 초록색 부분에 해당한다. 반면, 차량의 현재 RSU와 목적지를 담당하

는 RSU가 별개인 경우, 차량의 현재 RSU는 차량의 주차장 할당 요청을 클라우드로 전송하고, 클라우드가 목적지를 담당하는 RSU를 찾아 주차장 할당 요청을 의뢰하게 된다. 주차장 할당 요청을 받은 차량의 목적지 RSU는 담당 구역 내 주차장 중 가용한 최적의 주차장을 차량에게 할당한다. 그리고 그 결과를 다시 클라우드를 통해 차량의 현재 RSU로 전송하고, 차량의 현재 RSU가 차량에게 전송한다. 이는 그림 2에서 빨간색 부분에 해당한다. 운전자가 최종 선택을 하면, 위와 동일한 방법으로 목적지 RSU까지 전송되어 할당이 완료된다.

C. Tang, X et al.<sup>[6]</sup>은 차량이 직접 주차장 할당 요청을 진행하는 방안을 제안하였다. 그림 3과 같이 시스템은 차량, RSU, Fog node, 클라우드로 구성된다. 각 주차장에 설치된 Fog node는 주차장의 정보를 실시간으로 저장한다. 그리고 클라우드와 해당 구역을 담당하는 RSU에게 주기적인 비콘 메시지를 사용하여 주차장 정보를 전달한다. 따라서, 각 RSU는 담당하는 구역 내의 모든 주차장에 대한 정보를 유지하게 되고, 클라우드는 서비스 영역 전체의 모든 주차장 정보를 유지하게 된다. Fog node들의 주기적인 비콘 메시지로부터 담당 구역 내 주차장 정보를 수집한 RSU는 이를 주변 차량에게 역시 주기적인 비콘 메시지로 브로드캐스트한다. 차량은 이러한 RSU의 비콘 메시지로부터 적절한 주차장을 찾을 수 없는 경우, 주차장 할당 요청을 전송하지 않고 주차장을 할당 받지 않는다. 하지만 조건에 맞는 적절한 주차장이 있는 경우, 차량은 해당 주차장을 선택하고 담당 RSU를 통해 선택한 주차장의 Fog node로 주차장 할당 요청을 보낸다. [2], [5]에서 제안한 방안에서는 차량이 현재 위치에 관계없이 주차장 할당 요청을 보낼

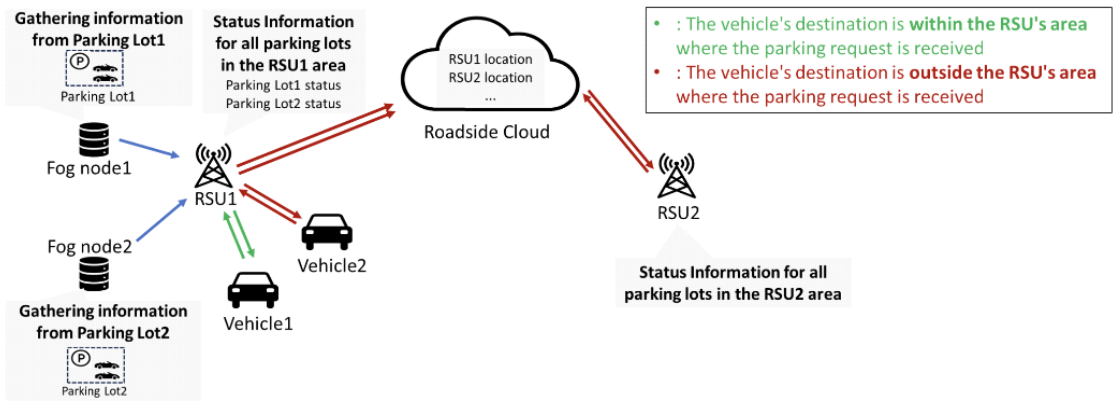


그림 2. O. Tran Thi Kim et al.<sup>[5]</sup>의 구성 요소  
Fig. 2. Components of O. Tran Thi Kim et al.<sup>[5]</sup>

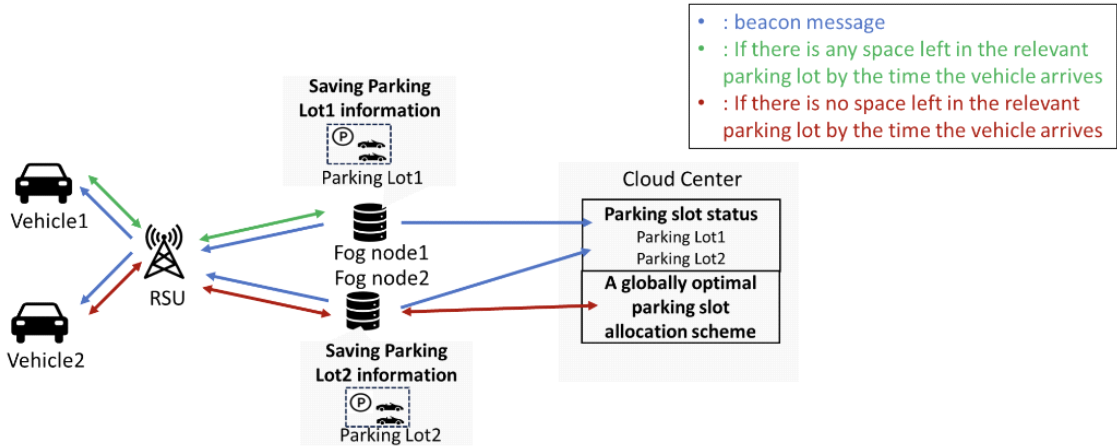


그림 3. C.Tang, X et al.<sup>[6]</sup>의 구성 요소  
 Fig. 3. Components of C.Tang, X et al.<sup>[6]</sup>

수 있었지만, [6]에서는 그림 8(d)와 같이 차량의 현재 RSU가 차량의 목적지 근처 RSU들 중 하나와 동일하게 되는 시점에 이르러야만 차량이 필요한 주차장 정보를 파악할 수 있기 때문에 주차장 할당 요청을 시작할 수 있다. 주차장 할당 요청을 받은 Fog node는 차량이 도착하기 전까지 가능한 주차장이 생길지를 확인한다. 가능한 주차장이 있는 경우에는 Fog node에서 바로 주차장을 할당하고, RSU를 통해 차량에게 할당 결과를 전달한다. 운전자의 최종 확인 과정도 Fog node와 이루어진다. 그림 3에서 초록색 부분이 이 과정을 보이고 있다. 하지만 차량으로부터 할당 요청을 받은 Fog node가 담당하는 주차장에 빈 공간이 없는 경우에는 Fog node는 클라우드에 주차장 할당 요청을 의뢰한다. 클라우드는 저장하고 있는 서비스 영역 내의 모든 주차장의 정보를 활용하여 할당을 진행하고, 그 결과를 처음 차량이 선택한 Fog node로 전송하여 RSU를 통해 차량이 결과를 받도록 한다. 이는 그림 3의 빨간색 부분에 해당한다. 이 경우, 운전자의 최종 확인은 주차장 할당 요청 때와 마찬가지로 클라우드까지 전송되고, 이를 받은 클라우드가 새롭게 선택된 Fog node에 이 사실을 알린다.

### III. MEC 기반의 스마트 주차 시스템

#### 3.1 구성 요소

제안하는 MEC 기반의 스마트 주차 시스템은, 그림 4와 같이 차량, 무선 기지국, MEC 서버로 구성된다. 무선 기지국의 경우, 제안하는 방안에서는 LTE를 사용하는 것으로 가정했기 때문에, eNodeB(eNB)로 구성되

어 있다. MEC 서버는 eNB에 위치할 수 있기 때문에<sup>[8]</sup>, 제안 방안에서는 하나의 eNB가 하나의 MEC 서버와 연동되어 있다고 가정하였다. 특별히, 차량의 현재 위치를 담당하는 MEC server를 Curr. MEC server라 지칭하고, 차량의 목적지를 중심으로 형성된 주차장 검색 구역을 담당하는 모든 MEC server들을 Dest. MEC servers라 부르기로 한다.

차량은 Global Positioning System(GPS)를 통해 차량의 현재 위치 정보를 수집한다고 가정하였다. 그리고 운전자의 주차 요청으로부터 차량의 목적지 위치 정보와 주차장에 대한 운전자의 선호 사항에 대한 정보를 수집한다. 여기에는 운전자가 선호하는 주차장 유형, 지불가능한 시간당 최대 주차 요금, 목적지에서 주차장까지의 최대 허용 거리 정보가 해당한다. 또한, 운전자는 이러한 세 가지 사항에 대해 선호도 가중치를 설정한다. 제안 방안에서 차량은 주차장을 할당 받고 싶을 때, 현재 연결된 eNB를 통해 Curr. MEC server에게 위의 정보가 포함된 주차장 할당 요청을 보낸다.

각 MEC 서버는 담당 구역 내에 있는 모든 주차장으로부터 주차장 정보를 수집한다. 여기에 해당하는 정보는 주차장 이름, 주차장 유형, 시간당 주차 요금, 주차장 위치, 남은 주차면 수 정보가 있다. 각 MEC 서버는 여기에 추가로 예약된 주차면 수 정보를 저장하고 있다. 이러한 주차장 정보는 MEC 서버 내에서 차량에 대해 주차장을 예약할 때와 차량이 주차장에 출입할 때마다 주차장에서 MEC 서버로 알려져 갱신된다. 또한, 각 MEC 서버는 자신을 제외한 다른 MEC 서버들의 위치 정보를 저장하고 있다. 차량으로부터 주차장 할당 요청을 받은 Curr. MEC server는 Dest. MEC servers에게

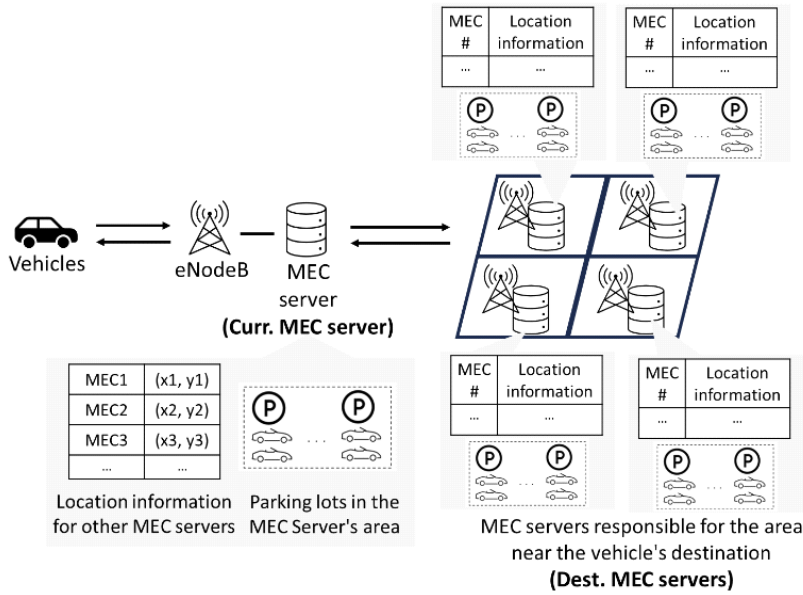


그림 4. 제안 방안의 구성 요소  
Fig. 4. Components of the system

주차장 할당 요청을 전달한다. Curr. MEC server로부터 주차장 할당 요청을 받은 Dest. MEC server는 자신이 담당하는 구역 내 주차장 중 가장 적합한 주차장을 선택해 응답 메시지로 보낸다. 담당 차량을 위해 Dest. MEC servers에게 주차장 할당 요청을 보낸 Curr. MEC

server는 이들 Dest. MEC servers의 응답 중 가장 적합한 주차장 정보를 선택해 차량에게 제공한다.

그림 5는 제안 방안 및 3가지 관련 연구에서 주차장 할당 요청을 처리할 때와 주차장 정보를 업데이트하는 과정에서, 데이터가 어떤 요소를 거쳐서 전달되는지에

(a) Proposed system		
Parking Request Processing	Vehicle → Curr. MEC server → Dest. MEC servers Dest. MEC servers → Curr. MEC server → Vehicle	
Parking Lot Information Management	Parking Lot → MEC server	
(b) MAPark <sup>[2]</sup>		
Parking Request Processing	Vehicle → Base Station → Cloud Cloud → Base Station → Vehicle	
Parking Lot Information Management	Parking Lot → Cloud	
(c) O. Tran Thi Kim et al. <sup>[5]</sup>		
Parking Request Processing	Vehicle → Dest. RSU Dest. RSU → Vehicle	
	Vehicle → Curr. RSU → Roadside Cloud → Dest. RSU Dest. RSU → Roadside Cloud → Curr. RSU → Vehicle	
Parking Lot Information Management	Parking Lot → Fog server → RSU	
(d) C. Tang, X et al. <sup>[6]</sup>		
Parking Request Processing	Vehicle → RSU → Fog node Fog node → RSU → Vehicle Vehicle → RSU → Fog node → Cloud Cloud → Fog node → RSU → Vehicle	
	Parking Lot Information Management	Parking Lot → Fog node → RSU, Cloud RSU → Vehicle

그림 5. 각 관련 연구 및 제안 방안에서의 데이터 전달 경로  
Fig. 5. Path of data transfer in each related study and proposed system

대해 정리한 것이다. 제안 방안은 MEC 서버들 사이의 협력을 통해 주차장 할당 요청을 처리한다. 반면, [2]에서는 항상 클라우드를 사용하고, [5]와 [6]에서는 상황에 따라 클라우드의 사용 여부를 결정하여 주차장 할당 요청을 처리한다. 주차장 정보의 업데이트 측면에서, 제안 방안은 MEC 서버가 담당 구역 내에 있는 주차장으로부터 주차장 정보를 받는다. [2]와 [6]에서는 클라우드가 전체 주차장 정보를 받아서 관리한다. 클라우드보다 MEC 서버가 더 차량과 가까운 곳에 위치하기 때문에, 제안 방안은 주차장 할당 요청 과정 및 주차장 정보 업데이트 과정에서의 네트워크 오버헤드와 지연을 줄일 수 있다. 또한 [6]에서는 주차장의 정보를 RSU와 차량에 주기적으로 전달하지만, 제안 방안에서는 주차장의 정보에 변화가 있을 때만 MEC 서버까지 전달하기 때문에 이 과정으로 인한 네트워크 오버헤드가 작다.

3.2 주차장 할당 과정

그림 6은 제안하는 방안에서 차량과 MEC server들이 주고받는 패킷의 종류와 각 패킷에 포함되는 정보를 보인 것이다.

모든 패킷은 제일 먼저 패킷의 종류를 포함하고 있다. 패킷의 종류가 0인 경우는 Request packet을 의미하고, 1인 경우는 Response packet, 2인 경우는 Assignment packet을 의미한다. 또한 각 패킷을 전송하는 주체에 따라, Car packet과 MEC packet으로 나뉜다.

Packet Type (0)	Vehicle ID	Current location of the vehicle(x, y)	Destination location of the vehicle (x, y)
Types of preferred parking lots	Maximum hourly parking charge payable	Maximum allowable distance from destination to the parking lot	
$w_1$	$w_2$	$w_3$	

(1) Request Car Packet (2) Request MEC Packet

Packet Type (1)	Vehicle ID	Parking Lot Name
Type of the parking lot	Parking fee	Location of the parking lot (x, y)
Number of spaces remaining in the parking lot	The score of parking lot by vehicle	

(3) Response MEC packet

Packet Type (1)	Vehicle ID	Parking Lot Name
Type of the parking lot	Parking fee	Location of the parking lot (x, y)
Number of spaces remaining in the parking lot		

(4) Response Car packet

Packet Type (2)	Vehicle ID
Parking Lot Name	Location of the parking lot (x, y)
Estimated Arrival Time	Estimated parking period

(5) Assignment Car Packet (6) Assignment MEC Packet

그림 6. 사용된 패킷의 종류  
Fig. 6. The types of protocol packets

Request Car와 Request MEC는 주차장 할당을 요청하는 패킷으로, 두 패킷은 동일한 내용을 포함하고 있다. 여기에 포함되는 것은 패킷의 종류, 차량의 ID, 차량의 현재 위치, 차량의 목적지 위치, 운전자가 선호하는 주차장의 종류와 이에 대한 가중치( $w_1$ ), 지불 가능한 시간당 최대 주차 요금과 이에 대한 가중치( $w_2$ ), 목적지에서 주차장까지의 최대 허용 거리와 이에 대한 가중치( $w_3$ )가 있다.

Response MEC와 Response Car는 주차장 할당 결과를 포함하고 있는 패킷이다. 그러므로, 여기에는 할당한 주차장의 이름, 종류, 요금, 위치와 남은 주차 공간의 수 정보가 포함된다. 단, 두 패킷의 정보 중 한 가지 차이점은 Response MEC에만 해당 주차장이 할당 받을 때 받은 점수 정보가 포함되어 있다는 점이다. Curr. MEC server에서는 이 점수를 기반으로 Dest. MEC server들이 제시한 주차장 중 가장 적합한 주차장 한 곳을 선정하여 차량에게 제시한다.

Assignment Car와 Assignment MEC는 차량이 할당 받은 주차장에 주차하기로 결정하여, 해당 주차장을 예약하기 위한 패킷이다. 이를 위해, 이 패킷에는 선택한 주차장의 이름, 위치, 주차장 도착까지의 예상 시간, 해당 주차장에서의 예상 주차 기간 정보를 포함한다.

그림 7은 주차장 할당 과정에서 차량과 MEC 서버들 간에 이들 패킷을 주고받는 과정을 보인 것이다. 먼저 차량은 주차장을 할당 받고 싶을 때, 현재 연결된 eNB를 통해 Curr. MEC server에게 Request Car를 전송한다.

Curr. MEC server에서는 Dest. MEC servers를 확인하기 위해, 차량으로부터 받은 Request Car에 명시되어 있는 차량의 목적지 위치, 목적지에서 주차장까지의 최대 허용 거리 정보를 활용한다. 차량의 목적지 위치를

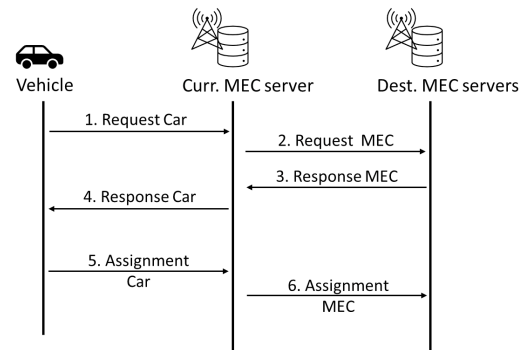


그림 7. 주차장 할당 과정  
Fig. 7. Parking lot allocation process

중심으로, 목적지에서 주차장까지의 최대 허용 거리를 반경으로 하는 구역을 주차장 검색 구역으로 설정한다. 그리고 Curr. MEC server는 다른 MEC 서버들의 위치 정보를 확인하여, 이 주차장 검색 구역의 일부분이라도 담당하고 있는 MEC 서버들은 모두 Dest. MEC servers로 설정한다. 이 과정은 식 (1)을 통해서 이루어진다.

$$\sqrt{(x_m - x_d)^2 + (y_m - y_d)^2} - r_d \leq r_m \quad (1)$$

식 (1)에서  $x_m$ 과  $y_m$ 은 MEC 서버의 위치,  $x_d$ 와  $y_d$ 는 차량 목적지의 위치다.  $r_m$ 은 MEC 서버가 담당하는 구역의 반경을 의미하고,  $r_d$ 는 목적지에서 주차장까지의 최대 허용 거리를 의미한다. Curr. MEC server는 이렇게 해서 결정된 모든 Dest. MEC servers에게 Request MEC를 전송한다.

그림 8은 제안 방안 및 각 관련 연구에서 주차장을 할당할 때, 고려하는 구역을 나타낸 것이다. 주차장을 할당할 때, 주차장 할당 성공률을 극대화하고, 할당의 만족도를 높이기 위해서는 차량의 목적지를 중심으로 운전자가 설정한 최대 반경 내에 속하는 주차장을 모두 고려해야 한다. 이후에서는 이를 “주차장 검색 구역”이라 부르기로 한다. 제안 방안은 식(1)을 통해 확인한 주차장 검색 구역 내의 Dest. MEC servers가 협력하여

주차장을 할당하기 때문에, 그림 8(a)와 같이 차량의 주차장 검색 구역을 모두 고려하여 그 가운데 최적의 주차장을 할당할 수 있다. [2]는 항상 서비스 영역 전체의 주차장 정보를 유지하고 있는 클라우드에서 주차장 할당이 이루어지기 때문에, 그림 8(b)와 같이 주차장 검색 구역을 모두 고려하여 최적의 주차장을 할당할 수 있다.

하지만, [5]는 전체 주차장 검색 구역 중 차량의 목적지를 담당하는 RSU가 담당하는 구역만 고려하기 때문에, 고려 대상이 되는 주차장이 제한적일 뿐 아니라, 그림 8(c)에서 보인 예와 같이 운전자가 가장 만족하는 주차장이 있는 구역이 고려 대상에 포함되지 못하는 경우가 발생할 수 있다.

[6]에서 차량은 주차장 검색 구역까지 접근한 후, 차량이 자신이 속한 RSU에게 주차장 할당을 요청한다. 따라서 [2], [5]의 방안들과 본 논문이 제안하는 방안에서는 차량이 어떤 위치에서라도 주차장 할당 요청을 시작할 수 있고, 차량의 현재 위치와 진행 방향에 상관없이 주차장 할당 시 고려되는 주차장 선택 구역의 범위가 동일했던 것에 반해, [6]에서 제안한 방안에서는 차량이 목적지로 접근하는 방향에 따라 주차장 할당 요청 시에 차량의 현재 위치를 담당하는 RSU가 달라지기 때문에 차량이 접근하는 방향에 따라, 그림 8(d)에서 보인 예와 같이 주차장 할당이 이루어지는 구역이 달라진다. 또한

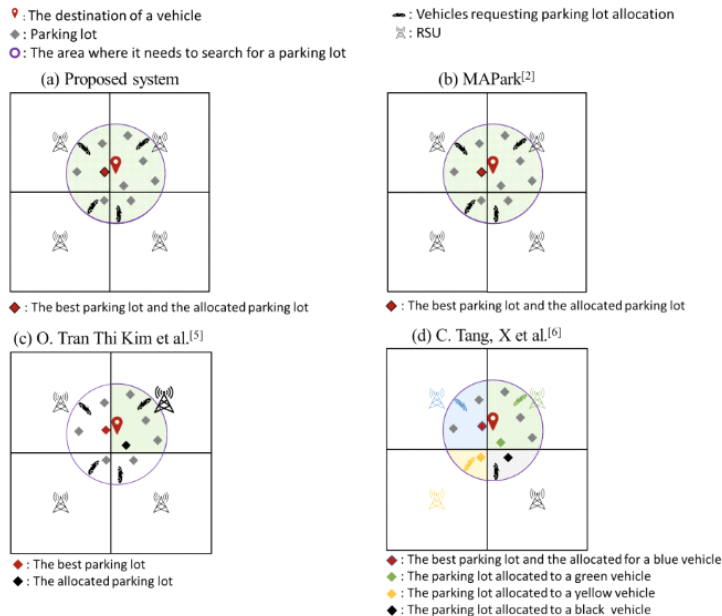


그림 8. 주차장 할당 시 고려하는 구역  
 Fig. 8. Areas to consider when allocating a parking lot

[5]의 방안과 마찬가지로 [6]에서도 전체 주차장 검색 구역 중 차량이 속한 RSU 한 개가 담당하는 구역 내에서만 주차장이 선택되기 때문에, 운전자가 가장 만족하는 주차장이 있는 구역이 고려 대상에 포함되지 못할 수 있다. 제안 방안은 [5], [6]과는 다르게 주차장 검색 구역을 모두 고려한다. 이는 [2]와 유사하지만, 클라우드를 사용한 [2]와는 달리 제안 방안은 클라우드보다 차량에 가까운 곳에 위치한 MEC servers만을 활용하면서 주차장 검색 구역을 모두 고려한다.

Request MEC를 받은 Dest. MEC Server에서는 Request를 보낸 차량에 대하여, 자신이 보유한 주차장 목록 내에서 가장 점수가 높은 주차장 한 개를 선정한다. 이때, 차량의 목적지를 기준으로, 목적지에서 주차장까지의 최대 허용 거리 이내에 있고 빈 주차면 수와 예약된 주차면 수를 고려했을 때 할당 가능한 주차면이 있으며, 지불 가능한 시간당 최대 주차 요금 이내의 요금에 해당하는 주차장에 대해서만 점수를 계산한다. 점수는 선호 주차장 종류, 시간당 주차 요금, 목적지에서 주차장까지의 거리를 고려하여 식 (2)와 같이 계산한다. 3개의 가중치의 합은 식 (3)과 같이 1로 유지한다. 각 요소의 가중치는 운전자가 직접 결정하고, 주차장에 대한 수요에 변동이 있어도 고정적인 주차장 정보를 활용하여, 관련 연구들보다 각 운전자를 중심으로 주차장을 할당한다.

$$Score = s_1 * w_1 + s_2 * w_2 + s_3 * w_3 \quad (2)$$

$$w_1 + w_2 + w_3 = 1 \quad (3)$$

$s_1$ 은 선호 주차장 종류에 대한 점수로, 식 (4)와 같이 점수를 계산하는 주차장이 운전자의 선호 주차장 종류와 일치하는 경우에는 값이 2를 갖고, 일치하지 않는 경우에는 값이 1을 갖는다.

$$s_1 \in \{1, 2\} \quad (4)$$

$s_2$ 는 시간당 주차 요금에 대한 점수로, 식 (5)와 같이 시간당 주차 요금이 적을수록 2에 가까운 값을 갖고, 운전자가 제시한 최댓값에 가까울수록 1에 가까운 값을 갖는다.  $s_2$ 는 식 (6)을 사용하여 구한다. 이를 위해, 각 주차장의 주차 요금에 대해 식 (7)을 사용하여 정규화 과정을 진행한다.  $f_p$ 는 점수를 계산하고자 하는 주차장의 주차요금,  $f_{d\_max}$ 는 운전자가 지불 가능한 시간당 최대 주차 요금이다.

$$1 \leq s_2 \leq 2 \quad (5)$$

$$s_2 = -n_2 + 2 \quad (6)$$

$$n_2 = \frac{f_p}{f_{d\_max}} \quad (7)$$

마지막으로  $s_3$ 은 목적지에서 주차장까지의 거리에 대한 점수로,  $s_2$ 와 마찬가지로 식 (8)과 같이 목적지에서 주차장까지의 거리가 가까울수록 2에 가까운 값을 갖고, 운전자가 설정한 최댓값에 가까울수록 1에 가까운 값을 갖는다.  $s_3$ 은 식 (9)를 사용하여 구한다. 이를 위해, 각 주차장에 대해 식 (10)을 사용해서 정규화를 진행한다.  $D_p$ 는 차량의 목적지와 점수를 계산하고자 하는 주차장 사이의 거리이고,  $D_{d\_max}$ 은 운전자가 허용한 주차장과 목적지 사이의 최대 거리이다.

$$1 \leq s_3 \leq 2 \quad (8)$$

$$s_3 = -n_3 + 2 \quad (9)$$

$$n_3 = \frac{D_p}{D_{d\_max}} \quad (10)$$

이렇게 구한  $s_1, s_2, s_3$ 를 식 (2)와 같이 각각의 가중치와 곱한 뒤, 합해서 각 주차장에 대한 점수를 구한다. 이때, 식 (3)과 식 (4), (5), (8)에 의해, 점수는 1에서 2사이의 값이 나온다.

각 Dest. MEC Server는 위와 같은 방법을 통해 가장 점수가 높은 주차장을 선정하여, 선택된 주차장에 대한 정보와 해당 점수를 Response MEC에 담아 Curr. MEC Server로 전송한다.

Curr. MEC Server는 Dest. MEC Servers로부터 Response MEC 받는 것을 일정 시간 동안 대기한다. 모든 Dest. MEC Servers로부터 Response MEC를 다 받았거나, 설정한 대기 시간이 만료된 경우, Curr. MEC Server는 Response MEC에 명시된 주차장 정보 중 가장 점수가 높은 주차장 한 개를 선정한다. 그리고 이 정보를 Response Car에 담아 주차장 할당을 요청했던 차량에게 전달한다.

Curr. MEC Server로부터 Response Car를 받은 차량은 운전자가 해당 주차장을 수락한다면, Assignment Car에 해당 주차장에 언제 도착할 것이고, 얼마나 머무를 것인지에 대한 정보를 담아 Curr. MEC Server로



전달한다.

Assignment MEC를 받은 Dest. MEC Server는 선택된 주차장을 예약하고, 예약된 주차면 수를 증가시킨다. 그리고 주차장으로부터 차량이 예상 도착 시간과 근접하게 주차장에 도착한 정보를 받은 후, 예약된 주차면 수와 빈 주차면 수를 감소시킨다.

#### IV. 실험

이 장에서는 NS-3 네트워크 시뮬레이터를 사용하여 제안하는 방안인 MEC 기반의 스마트 주차 시스템과 기존 방안인 MAPrak<sup>[2]</sup>, O. Tran Thi Kim et al.의 방안<sup>[5]</sup>, C. Tang, X et al.의 방안<sup>[6]</sup>의 성능을 비교하였다. 성능 비교는 주차장 할당 성능, 오버헤드, 지연 측면에서 진행하였다. 이를 위해, NS-3 네트워크 시뮬레이터에서 주차장 할당에 사용되는 패킷과 MEC 서버를 포함한 각 방안의 구성 요소를 구현하였고, LTE 모듈과 유선 네트워크 모듈 등을 활용하였다. 그리고, 이후의 설명에서는 각 방안에 대해 주차장을 할당하는 주체에 따라, 제안하는 방안은 MEC servers 방안, [2]는 Cloud 방안, [5]는 RSU 방안, [6]은 Client 방안이라 부르기로 한다.

표 1은 시뮬레이션 파라미터 값을 나타낸 것이다. 시뮬레이션에서 차량은 통신을 위해 LTE를 사용하였고, 6km × 6km의 전체 서비스 영역을 9개의 구역으로 균등하게 나누어, 각 구역마다 eNB와 연동된 RSU<sup>[9]</sup>를 배치하였다. 모든 방안에서 차량과 통신하는 기지국은 RSU로 통일하였다. 각 RSU가 담당하는 구역 내에 포함된 주차장의 수는 표 2와 같으며, 전체 지역에 있는 주차장의 수는 140개이다. 각 주차장은 7~25개의 주차

표 1. 실험 파라미터  
Table 1. Experimental parameters

Size of simulation space	6km x 6km
Total number of parking lots	140
Total number of parking area	2560 (Random values between 7 and 25 for each parking lot)
Vehicle speed	18~28km/h
Total number of parked vehicles	2100, 2800, 4200, 8400
Parking period per vehicle	Uniform Random Variable 30~90 minutes
Simulation time	5 hours

표 2. RSU별 주차장 수  
Table 2. Number of parking lots by RSU

RSU1	RSU2	RSU3	RSU4	RSU5	RSU6	RSU7	RSU8	RSU9
16	17	16	16	13	15	16	18	13

공간을 갖고 있다. 주차장의 종류에는 노상, 노외가 있으며, 주차장 요금은 서울시 공영주차장 안내 정보를 참고하여 각 주차장에 대해 5분당 요금으로 설정하였다.<sup>[13]</sup> 이러한 주차장 정보를 반영하여, 시뮬레이션에서 주차장 할당 요청을 생성할 때, 운전자가 선호하는 주차장 유형은 노상, 노외 중 하나를 선택한다. 지불 가능한 시간당 최대 주차 요금은 5분당 100원, 200원, 300원, 400원, 500원 중 하나를 선택한다. 목적지에서 주차장까지의 최대 허용 거리는 서울특별시 주차정보안내시스템을 참고하여, 500m, 750m, 1000m 중 하나를 선택한다.<sup>[14]</sup> 각 요소는 모두 무작위로 선택한다. 또한 각 요소에 대한 가중치는 0에서 1사이의 값을 가지면서, 세 가중치의 합이 1이 되도록 설정하였다. 주차장의 위치를 포함한 모든 상태정보는 모든 방안에서 동일하다. 각 차량의 시작 지점과 목적지는 실험하는 서비스 영역 내에서 임의로 선택되고, 각 차량은 시작 지점에서 목적지까지 직선으로 이동하도록 하였다. 그리고, 제안하는 MEC servers 방안, Cloud 방안, RSU 방안은 모두 같은 위치에서 주차장 할당 요청을 보내도록 하였다. Client 방안은 운전자가 허용하는 최대 반경 이내로 차량이 진입한 직후, 주차장 할당 요청을 진행하도록 하였다. Client 방안에서 만약 차량이 현재 RSU로부터 받은 주차장 목록 내에 운전자의 조건에 맞는 주차장이 없다면, 해당 차량은 주차장을 할당 받지 못하는 것으로 진행하였다. 이는 MEC servers 방안, Cloud 방안, RSU 방안과 동일하게 Client 방안에서도 차량이 주차장을 할당 받고자 하는 시점은 한 번으로 하기 위한 것이다. 또한 Client 방안은 주차장에서 RSU와 클라우드로 주차장 정보에 대한 비콘 메시지를 주기적으로 보내야 하는데, 클라우드로 주차장에서의 실제 정보의 차이가 발생하는 일이 거의 일어나지 않도록 전송 간격을 5초로 가정하였다. 차량의 평균 속도는 2022년 서울시 요일별 통행속도를 참고하여 설정했다.<sup>[10]</sup> 각 차량이 주차장에 주차하는 시간은 균등 분포를 사용하여, 30분에서 90분 사이의 시간을 모두 동일한 확률로 랜덤하게 선택하도록 설정했다. 이 주차 시간은 주차 안내 정보 시스템 구축 방안 연구에서 1회 평균주차시간 분포를 참고하여 설정한 것이다.<sup>[11]</sup> 비교 방안들이 제안된 각 논문은 주차장 할당을 통해 추구하고자 하는 목표가 다르기 때문에 서로 다른 주차장 할당 알고리즘을 사용하고 있지만,

본 논문에서 목표로 하는 운전자 만족도 충족 측면에서 비교가 이루어지도록 하기 위해 모든 비교 방안들에 대해 주차장 할당은 동일하게 본 논문에서 제안한 알고리즘을 적용하였다.

#### 4.1 주차장 할당 성능

각 방안에서 주차장 할당 성능을 비교하기 위해, 실험 시간 동안 각 차량들이 할당 받은 주차장의 점수를 구하여 비교하였다. 주차장 점수의 최고 점수는 2로, 2에 가까울수록 운전자가 원하는 주차장으로 할당한 것을 의미한다. 주차장 점수를 구할 때 최저 점수는 1로, 운전자가 제시한 조건에는 해당하지만 운전자의 만족도가 매우 낮은 주차장으로 할당한 것을 의미한다. 그리고 고려하는 모든 주차장에 빈 주차 공간이 없는 경우와 운전자가 제시한 조건에 해당하는 주차장이 없는 경우에는 할당을 받을 수 없기 때문에 이 경우는 0점으로 가정하였다.

그림 9는 주차 차량 수에 따라 각 방안에서 차량들이 할당 받은 주차장의 점수의 평균을 구하여 비교한 것이다. 모든 방안에서 차량 수가 증가했을 때, 평균 점수가 낮아지는 결과가 나타났는데, 이는 주차하는 차량 수가 증가하면서, 전체적으로 남은 주차 공간의 수가 감소하였기 때문이다. MEC servers 방안과 Cloud 방안은 동일한 평균 점수를 기록했다. Cloud 방안에서 서비스 전역의 모든 주차장 정보를 활용하여 최적의 주차장을 할당하지만 주차장을 할당할 때에는 어차피 검색 범위가 목적지 주변 지역으로 한정되기 때문에, 운전자가 원하는 최대 반경 내의 한정된 검색 범위를 담당하는 MEC 서버들이 협력해서 주차장을 할당을 하는 제안 방안의 주차장 할당 성능이 Cloud 방안과 동일하게 우수함을 보인 것이다. 결국 Cloud 방안과 MEC servers 방안은

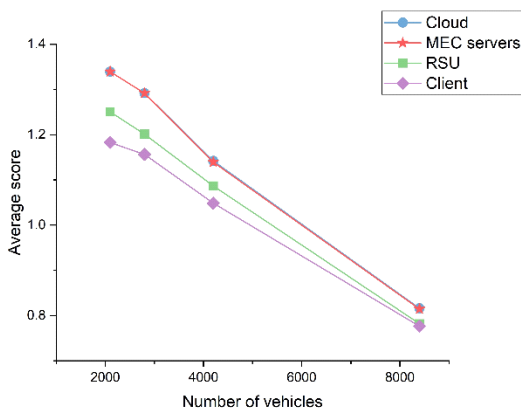


그림 9. 각 차량이 할당 받은 주차장 점수의 평균  
Fig. 9. Average parking lot score assigned to the vehicle

고려되어야 할 주차장들을 모두 고려하기 때문에 동일한 결과가 나왔다.

하지만 RSU 방안과 Client 방안은 Cloud 방안이나 제안하는 MEC servers 방안에 비해 더 낮은 평균 점수를 기록하였다. 이는 이 두 방안의 경우, 하나의 RSU 구역 내에서만 주차장 검색이 이루어지고, 그림 8에서 보는 바와 같이 더 적절한 주차장이 있는 구역이 검색에서 제외되는 경우가 있기 때문이다.

Client 방안은 RSU 방안에 비해서도 더 낮은 평균 점수를 기록했다. 이는 RSU 방안은 목적지 자체를 기준으로 목적지를 담당하는 RSU 구역 내에서 주차장 검색이 이루어지는데 반해, Client 방안의 경우, 차량이 목적지로부터 운전자가 허용하는 최대 반경 이내로 진입한 시점에 차량이 속한 RSU가 담당하는 구역에서 주차장 검색이 이루어지기 때문이다. 그런데, Client 방안의 경우 해당 RSU에서 주차장 할당이 실패했을 때는 클라우드에 주차장 할당을 의뢰하므로, 주차하려는 차량의 수가 늘어나면서 클라우드에 의한 주차장 할당이 더 많이 발생하게 되면서 RSU 방안과 Client 방안의 평균 점수 차이가 줄어들게 된다.

그림 10은 주차 차량 수가 증가함에 따라 각 방안에서 주차장을 할당 받지 못한 차량의 수를 비교한 것이다. 평균 점수와 마찬가지로 Cloud 방안과 MEC servers 방안은 RSU 방안과 Client 방안에 비해 주차장을 할당 받지 못한 차량의 수가 더 적었다. 또한, 평균 점수의 경우와 유사하게 주차장을 할당 받지 못하는 차량 수에서도 Client 방안이 RSU 방안보다 약간 좋지 않은 성능을 보였다. 이는 Client 방안에서 주차장을 할당 받고자 하는 시점에 차량이 현재 RSU로부터 받은 주차장

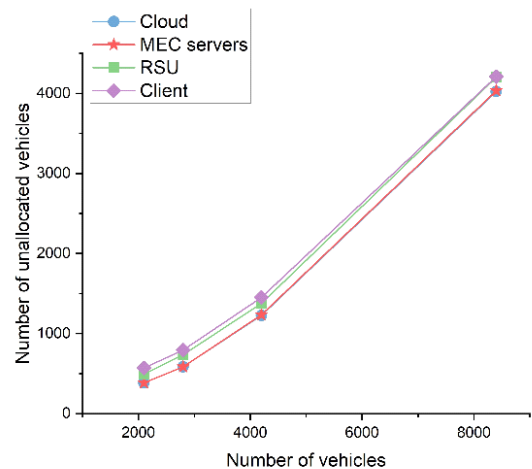


그림 10. 할당 받지 못한 차량의 수  
Fig. 10. Number of unallocated vehicles

목록 내에 운전자의 조건에 맞는 주차장이 없는 경우가 많았기 때문이다. 그러나, 평균 점수의 경우와는 달리 두 방안의 차이가 매우 적었고, 특히 차량의 수가 많은 경우는 RSU 방안과 Client 방안의 차이가 거의 없어지는 것을 볼 수 있는데, 이는 Client 방안에서는 차량의 RSU 구역 내에서 주차장 할당이 이루어지지 못하는 경우 클라우드에게 이를 의뢰하기 때문이다.

4.2 오버헤드 비교

각 구조에 대한 오버헤드를 비교하기 위해서, 패킷이 경유하는 경로를 차량과 RSU, RSU와 RSU, 주차장과 RSU, RSU와 클라우드, 주차장과 클라우드, 총 5가지의 구간으로 나눴다. 이들 5 종류의 구간은 각각 서로 다른 네트워크 오버헤드 및 지연을 발생시키기 때문이다. 이 실험에서 기존 방안의 주차장에 설치된 Fog node는 주차장으로 간주하고 진행하였다. 그리고 제안하는 방안에서 eNB와 MEC 서버가 하나로 연결되어 있다고 하였는데, 4장 실험에서는 eNB와 연동된 RSU로 가정했기 때문에 구간도 RSU와 RSU 구간으로 설정하였다.

5개 구간에 대해, 홉 수는 다음과 같이 가정했다. 먼저 차량과 RSU, 주차장과 RSU는 모두 RSU와 바로 통신하기 때문에 1홉으로 가정했다. 그리고 RSU와 RSU는 중간에 하나의 라우터가 있는 것으로 가정해서 2홉으로 설정했다. 마지막으로 RSU와 클라우드, 주차장과 클라우드의 홉 수를 설정하기 위해, Amazon Web Services(AWS)에서 공개한 서울 지역 클라우드의 Service API endpoint 주소로 Traceroute를 진행하였다. 이때, 결과로 20~25개의 노드를 통과하였기 때문에, 클라우드와 통신하는 두 구간은 20~25hop 범위에서 무작위로 선택하였다.

표 3은 주차 차량 수 2100대, 2800대, 4200대, 8400대에 대해, 각 구간별로 패킷 전송이 발생한 횟수를 비교한 것이고, 표 4는 각 구간별 홉 수를 반영하여 전체 네트워크에서의 패킷 오버헤드를 비교한 것이다. Client 방안은 주차장 정보에 대한 비콘 메시지를 주기적으로 클라우드, RSU, 차량에게 전송하기 때문에 모든 구간에서 패킷의 수가 다른 구조에 비해 상당히 많았다. Cloud 방안과 RSU 방안은 클라우드를 포함하는 구간을 경유하는 패킷의 양이 많지만, MEC servers 방안은 클라우드를 포함하는 구간에 발생하는 패킷이 전혀 없고 RSU 간에 주고받는 패킷이 많은 것을 확인할 수 있다. 클라우드와 패킷을 주고받는 구간은 다른 구간보다 경로 상에 노드가 더 많기 때문에, 전체적인 네트워크의 측면에서 오버헤드가 더 크다. 따라서 표 4에서 보는 바와 같이 결과적으로 네트워크 전체에서의 전송

표 3. 각 구간별 패킷 수  
Table 3. Number of packets per interval

(a) Number of vehicles - 2100

	MEC servers	Cloud	RSU	Client
Vehicle - RSU	5915	5915	5805	36993
RSU - RSU	10406	-	-	-
Parking lot - RSU	1518	-	1436	508593
RSU - Cloud	-	5915	10589	-
Parking lot - Cloud	-	1519	-	504000

(b) Number of vehicles - 2800

	MEC servers	Cloud	RSU	Client
Vehicle - RSU	7816	7816	7664	38421
RSU - RSU	13828	-	-	-
Parking lot - RSU	2120	-	1983	510021
RSU - Cloud	-	7816	13894	-
Parking lot - Cloud	-	2121	-	504000

(c) Number of vehicles - 4200

	MEC servers	Cloud	RSU	Client
Vehicle - RSU	11365	11372	11222	40652
RSU - RSU	20400	-	-	-
Parking lot - RSU	2974	-	2784	512252
RSU - Cloud	-	11372	20308	-
Parking lot - Cloud	-	2978	-	504010

(d) Number of vehicles - 8400

	MEC servers	Cloud	RSU	Client
Vehicle - RSU	21166	21178	20998	45004
RSU - RSU	39671	-	-	-
Parking lot - RSU	4898	-	4676	516604
RSU - Cloud	-	21178	37994	-
Parking lot - Cloud	-	4905	-	504058

표 4. 전체 패킷 오버헤드  
Table 4. Total packet overhead

	2100	2800	4200	8400
MEC servers	28245	37592	55139	105406
Cloud	173459	231504	334522	608027
RSU	245563	322528	471354	880424
Client	11886259	11889620	11893604	11901220

오버헤드는 클라우드를 포함하는 구간이 없는 MEC servers 방안에서 단연 가장 적음을 볼 수 있다. 그리고 Cloud 방안과 RSU 방안 중 RSU 방안이 클라우드와 주고받는 패킷의 양이 더 많은 것으로 나타났다. Cloud 방안에서는 현재 RSU와 클라우드 사이에서 직접 주차 할당 요청과 결과를 주고받기 때문에, RSU와 클라우드 구간의 1회 왕복만으로 주차 할당이 이루어지는 것에 반해, RSU 방안에서는 주차장 할당 요청을 한 차량의 현재 RSU와 목적지 RSU가 다른 경우, 두 개의 RSU 사이에 클라우드가 개입하게 되므로 차량의 현재 RSU는 클라우드로, 클라우드는 차량의 목적지 RSU로 주차장 할당 요청을 전송하고, 결과는 이 경로의 반대로 전송한다. 즉, RSU 방안에서는 클라우드가 개입되어야 하는 주차 요청의 경우에는 RSU와 클라우드 구간에서 2회 패킷 왕복이 있어야만 주차 할당이 이루어질 수 있다. 그런데, RSU 방안에 대한 실험에서 이와 같은 경우가 많이 발생했고, 차량의 수가 증가할수록 이러한 경우가 더 빈번히 발생하기 때문에 표3과 같은 결과가 나타났다.

그림 11은 주차 차량 수를 2100대~8400대로 증가시켜보면서 각 방안에서 주차 할당 과정에서의 평균 홉

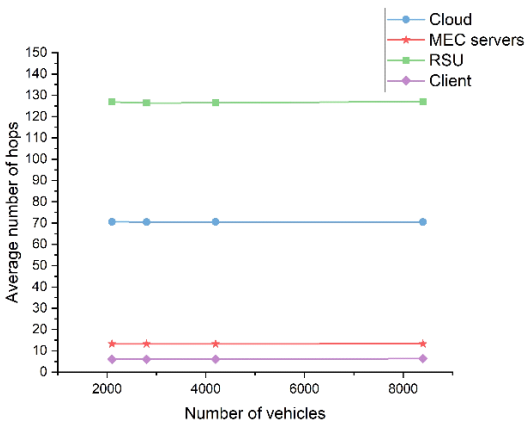


그림 11. 주차장 할당 과정에서의 평균 홉 수  
Fig. 11. Average number of hops in the parking allocation process

수만을 비교한 것이다. RSU 방안은 차량의 목적지 RSU를 찾기 위해 클라우드가 두 개의 RSU 사이에서 데이터를 중계하는 경우 클라우드가 2회 왕복이 요구되는데, 이와 같은 경우가 빈번하게 발생하기 때문에 평균 홉 수가 가장 많은 것으로 나타났다. Cloud 방안은 모든 주차장 할당을 클라우드에서 진행하기 때문에, MEC servers 방안과 Client 방안보다 평균 홉 수가 많았다. MEC servers 방안은 주차장 할당 과정에서 다른 MEC servers와 협력해야 하지만, Client 방안에서는 주로 차량과 RSU 간의 통신으로만 주차장 할당이 이루어지고 클라우드에서 할당이 진행된 경우가 전체 차량 중 적은 비율이었기 때문에, MEC servers 방안보다 더 적은 평균 홉 수를 보였다.

### 4.3 지연 비교

각 방안에서 차량이 주차 할당 요청을 하는 과정에서의 지연을 비교하기 위해, 주차장 할당 요청 및 응답 과정에서 패킷을 주고받는 데에 걸리는 시간에 대해서 실험하였다. 이를 위해 4.2에서 언급한 5개의 구간에 대해, 다음과 같이 딜레이를 설정하였다. 차량과 RSU, 주차장과 RSU 구간은 LTE를 사용하는 구간이다. 이때, LTE에서 사용하는 모드에 따라 차이가 있기는 하지만, 이를 모두 고려하여, Uplink의 지연은 4~8.5ms, Downlink의 지연은 4~5.2ms<sup>[12]</sup>로 가정했다. 유선 연결을 가정한 RSU와 RSU 구간의 단방향 지연은 1ms~2ms의 값으로 가정했다. RSU와 클라우드, 주차장과 클라우드 구간은 홉 수를 설정할 때 동일하게 AWS의 클라우드 주소로 ping을 전송하여 평균 지연을 구하였고, 이를 바탕으로 클라우드를 포함하는 구간의 단방향 지연을 13ms~17ms로 가정했다.

그림 12는 주차 차량 수 2100대~8400대에서 평균 지연을 비교한 것이다. 그림 11에서 평균 홉 수가 가장 많았던 RSU 방안이 평균 지연이 가장 긴 것으로 나타났다. 또한 그림 11과 동일하게 Cloud 방안이 두번째로 평균 지연이 긴 것으로 나타났다. 하지만 그림 11의 결과에서 MEC servers 방안이 Client 방안보다 평균 홉 수는 더 많았던 것에 반해, 평균 지연은 더 짧은 것으로 나타났다. 이는 MEC servers 방안은 다른 MEC servers와 협력할 때, 유선 연결인 RSU와 RSU 구간을 통해 패킷을 주고받지만, Client 방안은 클라우드를 사용하지 않는 경우에도 유선 연결보다 지연이 긴 LTE를 사용하는 구간을 통해 패킷을 주고받기 때문이다. 이로 인해 MEC servers 방안의 평균 지연이 Client 방안의 평균 지연보다 짧았다.

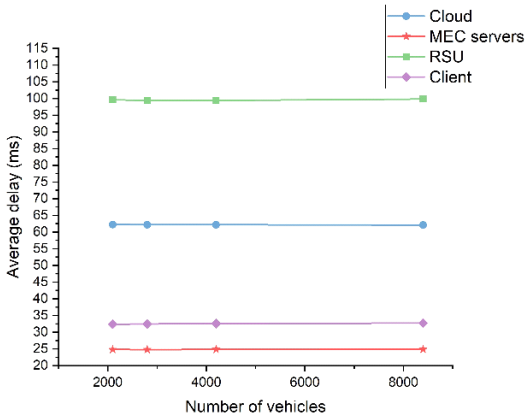


그림 12. 주차장 할당 과정에서의 평균 지연  
Fig. 12. Average delay in the parking allocation process

### V. 결 론

본 논문에서는 스마트 주차 시스템에서 기존보다 빠르면서, 정확한 주차장 할당을 하기 위해 MEC 기반의 스마트 주차 시스템을 제안하였다.

Cloud 방안은 항상 클라우드가 주차장을 할당하므로, 서비스 구역 내의 모든 주차장을 고려하지만, 네트워크 오버헤드가 크고 지연이 길다. RSU 방안과 Client 방안은 상황에 따라 클라우드를 사용하지 않고 빠르게 주차장을 할당할 수 있으나, 주차장 검색 구역의 일부만 고려하여 최적의 주차장을 찾지 못할 수 있다. Client 방안은 주기적인 비콘 메시지를 자주 전송하여 네트워크 오버헤드가 크다. 하지만 제안 방안은 MEC 서버들 사이의 통신을 통해 협력적인 할당을 진행하여, Cloud 방안처럼 주차장 검색 구역에 포함된 모든 주차장을 고려하므로 할당 만족도가 높다. 또한 제안하는 시스템은 지리적으로 클라우드보다 차량에 더 가까운 MEC 서버에서 주차장을 할당하므로, 작은 오버헤드와 짧은 지연을 달성할 수 있다.

향후 연구에서는 게임 이론 혹은 강화 학습을 사용하여, 동시에 발생한 주차장 할당 요청과 미래에 발생할 주차장 할당 요청을 모두 고려하여, 운전자의 만족도 범위 내에 해당하는 주변의 다른 주차장으로 분산시켜 더 많은 차량이 주차장을 할당받을 수 있도록 하는 방안을 연구하고자 한다.

### References

[1] J. Jang and T.-H. T. Gim, "A study on parking policy approach based on the social

costs of illegal parking," *J. Transport Res.*, vol. 24, no. 3, pp. 45-59, Sep. 2017.

(<http://dx.doi.org/10.34143/jtr.2017.24.3.45>)

[2] S. R. Rizvi, S. Zehra, and S. Olariu, "MAPark: A multi-agent auction-based parking system in internet of things," *IEEE Intell. Transport. Syst. Mag.*, vol. 13, no. 4, pp. 104-115, Winter 2021.

(<https://doi.org/10.1109/MITS.2019.2953524>)

[3] S. R. Rizvi, S. Zehra, and S. Olariu, "ASPIRE: An agent-oriented smart parking recommendation system for smart cities," *IEEE Intell. Transport. Syst. Mag.*, vol. 11, no. 4, pp. 48-61, Winter 2019.

(<https://doi.org/10.1109/MITS.2018.2876569>)

[4] M. Nam and H. Lee, "Design of an open platform-based smart parking system," *J. The IEIE*, vol. 56, no. 11, Nov. 2019,

(<https://doi.org/10.5573/jieie.2019.56.11.39>)

[5] O. T. T. Kim, N. H. Tran, C. Pham, T. LeAnh, M. T. Thai, and C. S. Hong, "Parking assignment: Minimizing parking expenses and balancing parking demand among multiple parking lots," *IEEE Trans. Automat. Sci. and Eng.*, vol. 17, no. 3, pp. 1320-1331, Jul. 2020.

(<https://doi.org/10.1109/TASE.2019.2948200>)

[6] C. Tang, X. Wei, C. Zhu, W. Chen, and J. J. P. C. Rodrigues, "Towards smart parking based on fog computing," *IEEE Access*, vol. 6, pp. 70172-70185, 2018.

(<https://doi.org/10.1109/ACCESS.2018.2880972>)

[7] K. M. Lee and M. J. Lee, "The cooperative parking lot allocation system reflecting the scope of parking lot information utilization in the MEC environment," in *Proc. Symp. KICS*, pp. 568-569, Pyeongchang, Korea, Jan. 2024.

(<https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11737168>)

[8] S. G. Kim and J. D. Park, "Status of mobile edge computing technology towards 5G era," *Electr. and Telecommunications Trends*, vol. 31, no. 1, pp. 25-35, Feb. 2016.

(<https://doi.org/10.22648/ETRI.2016.J.310103>)

[9] S.-Y. Lien, C.-C. Chien, G. S.-T. Liu, H.-L.

Tsai, R. Li, and Y. J. Wang, "Enhanced LTE device-to-device proximity services," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 174-182, Dec. 2016.

(<https://doi.org/10.1109/MCOM.2016.1500670> CM)

- [10] Seoul Metropolitan Government, *Traffic Speed Report of Seoul Metropolitan Government* (2022), Retrieved Nov. 15, 2023, from [https://topis.seoul.go.kr/refRoom/openRefRoom\\_1.do](https://topis.seoul.go.kr/refRoom/openRefRoom_1.do)
- [11] M. Kim and M. Park, "Establishing the methods of parking guidance and information systems(PGI)," *Basic Research on Research Reports*, vol. 2003, no. 0, pp. 1-129, Dec. 2003.  
(<https://kiss.kstudy.com/Detail/Ar?key=4007741>)
- [12] Y. Xu, *Latency and bandwidth analysis of LTE for a smart grid*(2011), Retrieved Nov. 22, 2023, from <https://www.diva-portal.org/smash/get/diva2:565509/FULLTEXT01.pdf>
- [13] Seoul, *Information on Public Parking Lot in Seoul* (2023), Retrieved Nov. 15, 2023, from <https://data.seoul.go.kr/dataList/OA-13122/S/1/datasetView.do>
- [14] Seoul, *Seoul Parking Information System*, Retrieved Nov. 15, 2023, from <https://parking.seoul.go.kr/>

이 경 민 (Kyoung-min Lee)



2022년 2월 : 이화여자대학교 컴퓨터공학과 졸업

2024년 2월 : 이화여자대학교 컴퓨터공학과 석사

<관심분야> Wireless Mobile Network, Mobile Edge Computing, VANET, FANET

[ORCID:0009-0005-6974-5873]

이 미 정 (Mee-jeong Lee)



1987년 : 이화여자대학교 전자계산학과 졸업

1989년 : University of North Carolina at Chapel Hill 컴퓨터학과 석사

1994년 : University of North Carolina at Chapel Hill 컴퓨터학과 박사

1994년~현재 : 이화여자대학교 컴퓨터공학과 교수  
<관심분야> FANET, MANET, VANET, Task Offloading, Wireless Mobile Networks, Qos routing, Internet Traffic Engineering

[ORCID:0000-0001-6968-8817]