

모바일 엣지 환경에서 커넥티드카 실시간성 저해의 상관 관계 분석[†]

장준범, 최희석, 유현창
고려대학교 컴퓨터전파통신공학과
e-mail : {jb0726, hsrangken, yuhc}@korea.ac.kr

Correlation Analysis of Connected Car Realtime Inhibition In Mobile Edge Computing Environment

JuneBeom Jang, HeeSeok Choi, HeonChang Yu
Dept. of Radio Communication, Korea University

요 약

커넥티드카는 네트워크에 연결된 자동차가 다른 자동차 및 도로 인프라뿐만 아니라 스마트 디바이스와 통신하고 여러 소스로부터 실시간 데이터를 수집하여 다양한 서비스를 제공하는 것이다. 커넥티드카의 등장으로 인해서 자동차와 클라우드 서비스의 결합이 빠르게 진행되고 있으나 자동차 데이터 중 실시간 처리가 필수인 데이터가 많다는 특성이 있다. 그러므로 멀리 떨어진 중앙 집중식 서버에서 컴퓨팅을 하는 클라우드 컴퓨팅보다 최근 이슈가 되고 있는 디바이스와 가까운 가장자리에 위치한 서버에서 컴퓨팅을 하는 엣지 컴퓨팅이 커넥티드카의 실시간성을 보장하는 기술로 많은 관심을 받고 있다. 본 논문에서는 기존의 엣지 컴퓨팅과는 달리, 이동성이 있는 모바일 엣지 컴퓨팅(MEC) 환경에서 실시간 처리를 저해하는 요소를 찾아 원인을 분석하고 평가해 문제점을 해결하고자 한다. 먼저, MEC 환경을 구축한 후 오픈 소스 시뮬레이터인 Edge Cloudsim 에 적용시켜 시뮬레이션을 한다. 실험 결과 MEC 환경에서 실시간 처리를 저해하는 원인은 모바일 디바이스의 태스크가 오프로딩 되거나 응답을 받기 전 WLAN 의 범위를 벗어났을 때 Task Failure 가 발생하기 때문임이 증명되었다.

1. 서론

최근 사물 간 상호통신을 바탕으로 자동차가 일상 생활의 허브 역할을 하는 커넥티드카의 관심이 증가하고 있다. 가트너는 2020년에 전 세계 2억 5000만대 이상의 차량이 무선 네트워크에 연결될 것이라고 전망했다[1]. 하지만 커넥티드카의 가장 큰 이슈는 자동차에서 발생하는 데이터의 실시간 처리이다. 이러한 이슈를 극복하기 위해서 클라우드 컴퓨팅을 적용하였지만 중앙 집중식 서버 기반 클라우드 컴퓨팅은 서버와 클라이언트간 발생하는 지연시간으로 인해 실시간 처리에 적합하지 않을 수 있다. 문제 해결을 위해 디바이스 주변에 위치한 서버에서 컴퓨팅을 하는 엣지 컴퓨팅의 등장으로 실시간성을 보장하는 이슈는 일부 해결이 되었다.

자동차용 엣지 컴퓨팅 환경을 구축하기 위해 ETSI 에서 기지국에 엣지 서버를 두는 시나리오를 제시했

으나 전체 기지국에 엣지 서버를 구축하는 비용이 너무 많이 소요되어 저비용의 엣지 서버의 필요성이 대두되었다. 또한 엣지 컴퓨팅 환경에서도 미미하게 실시간성을 저해하는 요소가 있을 수 있다.

본 논문에서는 기존의 엣지 컴퓨팅과는 다른 모바일 디바이스에 소프트웨어적으로 엣지 서버를 구현한 새로운 모델인 MEC 환경에서 임의적으로 엣지 서버에 태스크를 할당하고 극단적인 상황까지 태스크를 증가시켜 실험을 진행하여 실시간성을 저해하는 원인을 찾는다.

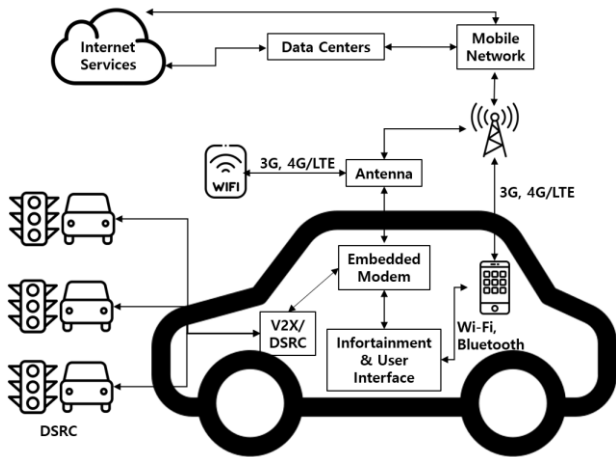
2. 관련 연구

커넥티드카는 인터넷에 액세스할 수 있는 차량으로서 다른 차량 및 도로 인프라뿐만 아니라 스마트 장치와 통신하고 여러 소스로부터 실시간 데이터를 수집하여 다양한 서비스를 제공하는 개념이다. (그림 1)

[†] “본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터지원사업의 연구결과로 수행되었음” (IITP-2018-0-01405)

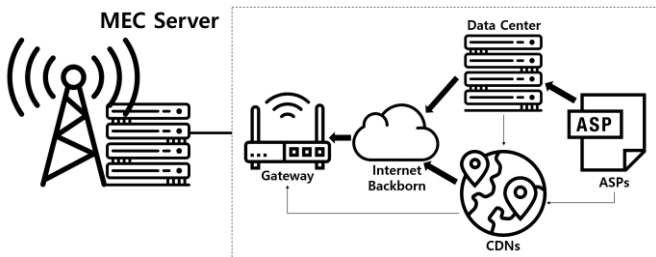
[‡] 이 논문은 2019년도 정보(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2018-0-00480)

은 커넥티드카의 아키텍처를 보여준다. 커넥티드카는 차량 및 도로 인프라와 Vehicle to Everything(V2X) 통신 기능을 통해 주변 차량들의 위치 및 속도 정보와 현재 차량의 상태 등을 공유한다. 커넥티드카의 장점은 첫 번째로 연결된 차량 간의 정보 공유를 통해 운전자에게 장애물, 교통량 등 중요한 정보를 제공하여 안전성을 높일 수 있다. 두 번째로 교통 정보를 실시간으로 제공받아 경로 최적화를 가능하게 하여 목적지에 더 빨리 도착하며 연료 소비를 줄이는데 기여하므로 시간과 비용이 절약된다. 세 번째로 교통 혼잡을 줄이고 연료 효율성을 높이므로 대기 질을 개선하고 탄소 배출량을 줄일 수 있으므로 환경에 이로운 영향을 준다[2].



(그림 1) 커넥티드카 아키텍처

MEC 는 네트워크 가장자리에 컴퓨팅, 분석, 저장 용량을 지원하는 네트워크 아키텍처다. 특히 IoT 애플리케이션에 성능상 이점을 제공할 수 있다. (그림 2) 는 MEC 의 아키텍처이다. MEC 의 특성으로는 첫 번째로 Proximity 가 있다. 모바일 이용자와 근접한 위치에서 서비스를 제공함에 따라 네트워크 레이턴시를 줄이고 Backhaul 의 대역폭 등 자원 사용을 줄인다. 두 번째로 Location Awareness 이다. 엣지 서버가 지역 단위로 분산되어 있어 지역 단위 서비스 제공이 가능하며 지역 특화 서비스 제공이 가능하다. 세 번째로 High QoE 이다. Base station 에서 수집 가능한 Wireless 네트워크의 상황에 따른 송출 스트림 품질 조절 및 콘텐츠 캐싱을 통한 QoE 의 향상이다[3].



(그림 2) MEC 아키텍처

3. 실험 환경

본 연구에서 기반하는 MEC 실험 환경은 사용자 스마트 디바이스에 소프트웨어로 구현된 엣지 서버가 위치한다. 또한 실시간성을 저해하는 원인을 찾기 위해 오픈소스 시뮬레이터인 Edge CloudSim 을 사용하여 시뮬레이션을 한다. Edge CloudSim 에서 사용된 파라미터와 그에 대한 값은 <표 1>과 같다. 모든 장소는 모두 WiFi 액세스 포인트가 존재하고 스마트 디바이스는 WLAN 에 연결된다고 가정한다. 또한 엣지에서 중앙 클라우드로 태스크나 결과가 오프로드 될 때는 WAN 연결이 사용된다. 어플리케이션은 의도적으로 로드를 발생시키는 컴퓨테이션 어플리케이션과 자동차에 정보를 제공하는 인포테인먼트 어플리케이션을 동시에 실행시켜 실험 환경을 구성하였다.

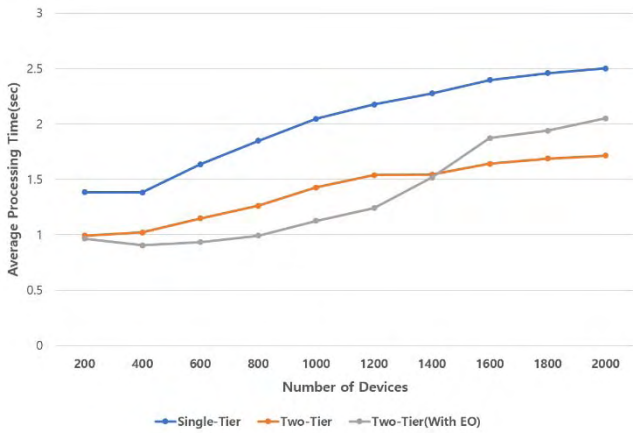
<표 1> Edge CloudSim Parameters.

Parameter	Value
Simulation Time(hour)	30
Minimum Number of Smart Devices	200
Maximum Number of Smart Devices	2000
Mobile Device Counter Size	200
Number of Host on Cloud Datacenter	1
WLAN bandwidth	200
WAN bandwidth	15

4. 실험 및 분석

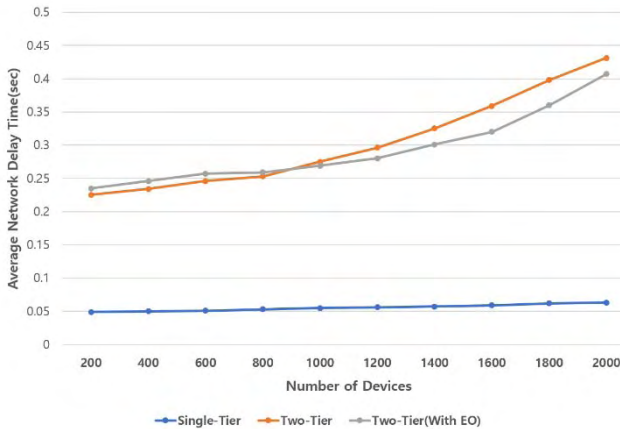
본 연구에서는 Edge CloudSim 의 세 가지 아키텍처 Single-Tier, Two-Tier, Two-Tier with Edge Orchestrator(EO) 로 시뮬레이션을 진행한다. Single-Tier 는 스마트 디바이스가 하나의 엣지 서버를 사용하는 아키텍처이다. Two-Tier 는 스마트 디바이스가 wide area network(WAN) 연결을 사용하여 중앙 클라우드에 태스크를 오프로딩하는 아키텍처이다. Two-Tier with EO 는 근접한 여러 개의 엣지에 태스크를 오프로딩하는 아키텍처이다. 스마트 디바이스 개수를 처음 200 개부터 시작하여 200 개씩 추가하여 마지막엔 2000 개까지 측정하여 스마트 디바이스 개수 당 성능을 시뮬레이션한다.

(그림 3)는 각각 스마트 디바이스 개수 당 평균 태스크 처리 시간이다. 세 가지 아키텍처 모두 스마트 디바이스의 개수가 증가하면 서비스 시간과 처리 시간이 증가한다. 서비스 시간과 처리 시간이 늘어날수록 그만큼 응답 시간이 길어지게 되며 짧은 응답 시간이 요구되는 커넥티드카의 실시간성을 저해하는 원인이 될 수 있으나 그 영향은 미미한 것으로 볼 수 있다.



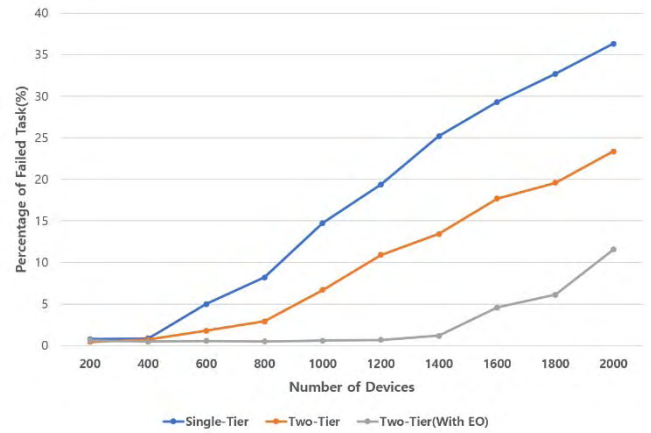
(그림 3) 스마트 디바이스 개수 당 평균 처리 시간

(그림 4)은 스마트 디바이스 개수 당 평균 네트워크 지연 시간이다. Single-Tier 아키텍처는 하나의 엣지만 사용하므로 네트워크 지연에 미미한 변화를 보이지만 Two-Tier, Two-Tier(EO) 아키텍처는 스마트 디바이스의 개수가 증가할수록 네트워크 지연시간이 높아진다. 이는 태스크 오프로딩으로 인한 네트워크 지연시간이 발생하기 때문이다. 하지만 네트워크 지연시간이 매우 낮기 때문에 실시간성에 미치는 영향은 거의 없다.



(그림 4) 스마트 디바이스 개수 당 평균 네트워크 지연시간

(그림 5)은 스마트 디바이스 개수 당 태스크 실패 비율을 보여준다. 태스크 실패란 태스크를 오프로딩했을 때 그에 대한 응답이 오지 않는 경우를 뜻한다. 태스크 실패 비율은 전체 태스크에서 실패한 태스크의 비율이다. 실시간성이 중요한 커넥티드카 환경에서 태스크 실패 비율이 높아 정상적인 응답을 받지 못해 자동차를 제대로 제어할 수 없는 것은 치명적인 문제이다. Single-Tier 는 최대 36 퍼센트, Two-Tier 아키텍처는 최대 23 퍼센트의 태스크 실패 비율이 나타난다. 두 아키텍처는 스마트 디바이스 개수가 증가할수록 선형적으로 증가하는 것을 볼 수 있다. 하지만 다양한 엣지 서버에 태스크를 분배하여 처리하는 Two-Tier(EO) 아키텍처에서는 태스크 실패 비율이 최대 11 퍼센트로 현저하게 낮은 것을 볼 수 있다.



(그림 5) 스마트 디바이스 개수 당 태스크 실패 비율

시험결과 (그림 3), (그림 4), (그림 5)를 토대로 스마트 디바이스의 개수가 늘어날수록 평균 서비스 시간, 평균 처리 시간, 태스크 실패 비율이 증가하지만 그 중 실시간성을 저해하는 큰 요소는 태스크 실패 비율인 것을 발견했다.

5. 결론

본 논문은 MEC 환경에서 커넥티드카의 실시간성을 저해하는 원인은 적절하지 못한 태스크 오프로딩으로 인한 태스크 실패가 가장 치명적이라는 것을 시뮬레이션을 통해 증명하였다. 그러므로 적절한 태스크 오프로딩 기법을 적용하여 태스크 실패 비율을 줄여 실시간성을 보장하는 것이 가장 중요하다. 태스크 오프로딩으로 인해서 네트워크 지연시간이 미미하게 증가하겠지만 커넥티드카에서 가장 중요한 것은 엔드 유저와 클라이언트 간 통신이 단절되지 않는 것이다. 향후 연구는 MEC 와 커넥티드카의 환경을 실제로 구현하여 연구해서 네트워크 지연시간의 증가를 최대한 줄이고 태스크 실패 비율을 낮추도록 개선할 것이다.

참고문헌

- [1] "Gartner Says by 2020, a Quarter Billion Connected Vehicles Will Enable New InVehicle Services and Automated Driving Capabilities," Gartner, 26 Jan. 2015.
- [2] COPPOLA, Riccardo; MORISIO, Maurizio. Connected car: technologies, issues, future trends. *ACM Computing Surveys (CSUR)*, 2016, 49.3: 46.
- [3] MAO, Yuyi, et al. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 2017, 19.4: 2322-2358.
- [4] SONMEZ, Cagatay; OZGOVDE, Atay; ERSOY, Cem. EdgeCloudSim: An environment for performance evaluation of Edge Computing systems. *Transactions on Emerging Telecommunications Technologies*, 2018, 29.11: e3493.