엣지 컴퓨팅 기반 무인 마켓 사례 연구: 자원 분배 효율성 극대화*

박지훈 [1], 류형오 [1], 김경률 [1], 김세화 [1]
[1] 한국외국어대학교 정보통신공학과
e-mail : ggoowlgns@gmail.com, 951010v@hanmail.net, kyoung831@naver.com, ksaehwa@hufs.ac.kr

Edge Computing-Based Unmanned Market Case Study: Maximizing Resource Distribution

Ji-Hoon Park[1], Hyeong-Oh Ryu[1], KyoungRul Kim[1], and Saehwa Kim[1]
[1]Dept of Information Communications Engineering, Hankuk Universitiy of Foreign Studies

요 약

본 논문에서는 엣지 컴퓨팅을 무인 마켓에 도입하여 엣지 컴퓨팅의 효율성을 확인하고, 로컬 네트워크의 효율적인 대역폭 할당을 위한 두 가지 방법을 제안한다. 무인 마켓과 같이 엄청난 양의 데이터를 필요로 하고 만들어내는 서비스에서는 데이터들을 클라우드로 전송하여 소비자가 불편함을 느끼지 못하도록 빠르게 처리하는 것은 불가능에 가깝다. 그래서 우리는 Amazon Go 를 벤치마킹한 무인 마켓에 엣지 컴퓨팅을 도입하여 이를 구현한다. 그리고 구현한 시스템에서 엣지 컴퓨팅 외에 클라우드 컴퓨팅, 모바일 장치를 적용하여 처리할 때의 응답 시간을 분석하여 엣지 컴퓨팅의 높은 성능을 확인한다. 또한, 구현한 무인 마켓에서 데이터 전송의 효율성을 더욱 높이기 위해 카메라 단위와 매대 단위의 대역폭 할당 기법을 제안한다. 카메라 단위로는 모션 인식기술을 활용하여 움직임이 감지될 때만 각 이미지 프로세스에서 요구되는 고해상도로 송신하는 기법을 제안한다. 매대 단위로는 네트워크에서 수용 가능한 대역폭 임계치에 도달하지 못하게 하기 위해 매대 별 우선순위에 따른 대역폭 할당 스케줄링 기법을 제안한다. 그 결과로 평균 소모대역폭과 최대 소모대역폭을 비교하여 제안한 두 가지 기법이 기존의 방법에 비해 성능을 향상시키는 것을 보인다.

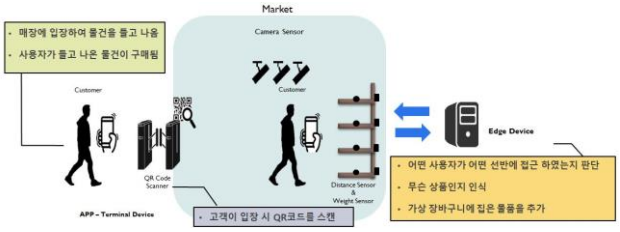
1. 서론

최근 AWS (Amazon Web Service) Greengrass, Microsoft Azure IoT edge, Google edge TPU 와 같은 엣지 컴퓨팅을 도입한 사례가 많아지고 있다. 이는 클라우드 컴퓨팅의 단점 때문인데, 클라우드 컴퓨팅은 클라우드와 거리에 따른 네트워크 지연이 크기 때문에 응답 시간에 민감한 어플리케이션에 문제가 될 수 있다. 뿐만 아니라 민감한 정보를 다루는 어플리케이션의 경우 다른 사람이 볼 수 있기 때문에 클라우드 컴퓨팅을 사용할 수 없다. 또한, 대량의 데이터를 생성하는 어플리케이션은 높은 비용을 발생시킬 수 있어 클라우드 컴퓨팅이 효율적이지 않다.

이와 같은 문제를 해결하기 위해 데이터 처리 능력을 사용자의 단말 장치들과 가까운 컴퓨팅 디바이스에 위치시키는 엣지 컴퓨팅이 생겨났다.[1] 엣지 컴퓨팅은 사용자의 단말 장치와 가까운 곳에 엣지 디바이스를 둠으로써 오프로딩할 때 거리상 멀리 있는 클라우드보다 응답시간이 빠르다. 그리고 엣지 디바이스로 오프로딩하여 하나의 사용자 단말 장치에서 얻은 정보로는 알 수 없는 정보들을 도출해 낼 수 있다.[2] 또한, 엣지 디바이스에서 데이터를 처리하여 클라우드로 데이터를 보내기 때문에 민감한 정보를 숨길 수 있으며, 대량의 데이터를 클라우드로 보내는 것을 막을 수 있다. 마지막으로 엣지 디바이스는 자체적으로 처리할 수 있는 능력이 있기 때문에 클라우드 컴퓨팅과 달리 외부 네트워크가 접근할 수 없는 상태일 경우에도 정상적으로 서비스를 제공할 수 있다.[3]

우리는 Amazon 사의 Amazon Go 를 벤치마킹한 무인 마켓을 구현하기 위해 엣지 컴퓨팅을 사용하였다. 이는 무인마켓에서 요구되는 기술인 얼굴 인식, 물체 감지 등이 막대한 데이터 처리량을 요구하기 때문이다. 그리고 엣지 컴퓨팅의 효율성을 확인하기 위해 무인마켓 데이터를 엣지 디바이스, 터미널 디바이스, 클라우드에 오프로딩하는 것을 비교하여 엣지 컴퓨팅의 효율성을 확인한다.[4] 클라우드 컴퓨팅은

* 이 논문은 2017 년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2017R1A2B1001824).



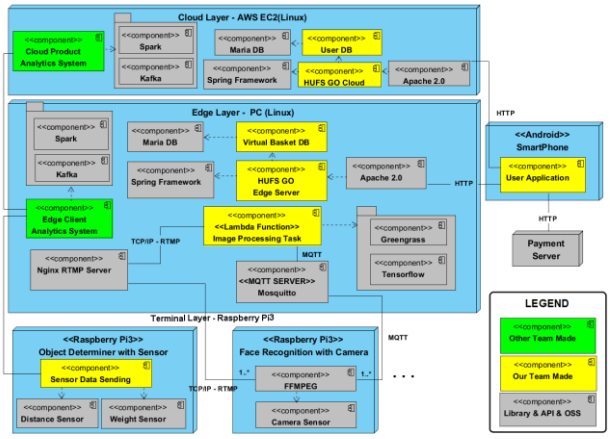
(그림 1) 무인 마켓 시나리오

AWS 를 사용하고, 엣지 컴퓨팅을 위한 소프트웨어는 AWS 클라우드 기능을 엣지 디바이스로 확장하는 AWS Greengrass 를 사용하였다.[5] 또한, 본 논문에서는 엣지 컴퓨팅의 효율성을 높이기 위해 두 가지 새로운 자원 분배 방법을 제시하고, 이를 기존의 방법과 비교 분석한다.

본 논문의 나머지 부분은 다음과 같이 구성되어 있다. 2 장에서 엣지 컴퓨팅 기반의 무인 마켓의 시나리오와 구성을 설명하고, 3 장에서 엣지 컴퓨팅 환경 내에서 효율적으로 자원을 분배하는 방법을 제시한다. 4 장에서는 엣지 컴퓨팅의 효율성을 확인하고 자원 분배 방법에 대한 실험과 그에 대한 결과를 설명한다. 마지막으로 5 장에서 결론을 맺는다.

2. 엣지 컴퓨팅 기반 무인 마켓

본 연구에서 구현한 엣지 컴퓨팅 기반 무인 마켓의 시나리오는 그림 1 과 같이 고객이 자신의 핸드폰 어플리케이션의 QR 코드를 찍고 마켓에 입장하여 매대에 있는 상품을 선택하면 얼굴인식, 물체감지, 센서 퓨전을 통하여 고객의 가상 장바구니에 선택된 상품을 담게 된다. 그리고 쇼핑이 끝난 고객은 나가면서 QR 코드를 다시 찍으면 자동 결제되는 시나리오로 수행된다. 그리고 그림 2 와 같이 웹 서버는 엣지, 클라우드 서버 모두에 각각 MariaDB 와 Spring 프레임워크로 구현하여 Apache2.0 으로 서비스하였다. 그리고 영상 정보를 위해 카메라(C922)를 연결한 라즈베리파이에서 FFMPEG 라이브러리를 활용하여 실시간 촬영 영상을 RTMP 로 멀티미디어 서버에 송신하였다. 멀티미디어 서버는 Nginx-rtmp 모듈을 활용한 Nginx 서버로 구현하여 영상들을 수신하였고. 수신된 각 영상별로 엣지 디바이스에서 텐서플로우로 구현한 Faster R-CNN 모델을 사용하여 이미지 프로세스를 실시하였다. 사용자 서비스용 어플리케이션은 Android Native App 으로 구현하였다. 엣지 디바이스와 라즈베리파이 그리고 고객 핸드폰은 모두 같은 NAT 환경 아래 있다고 가정하였다.



(그림 2) 무인 마켓 배치 다이어그램

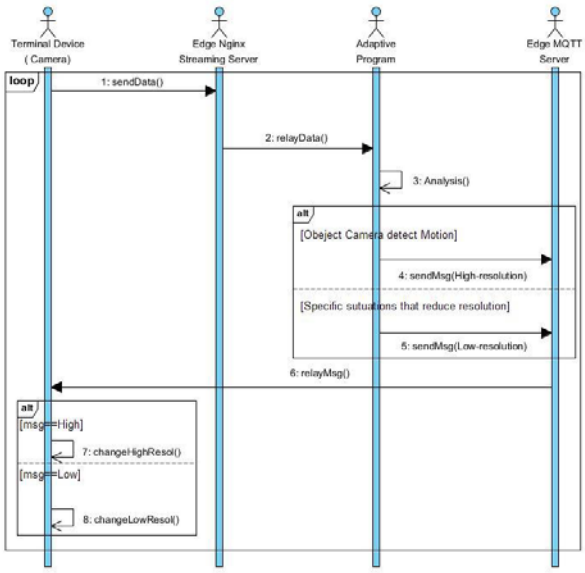
3. 엣지 컴퓨팅 효율화를 위한 자원 분배 기법

본 논문에서는 엣지 컴퓨팅의 효율성을 높이기 위해 모션 탐지 기반 대역폭 할당과 다중 스트림 스케줄링을 제안한다. 모션 탐지 기반 대역폭 할당은 무인 마켓 매대에 고객이 없어 해당 매대 카메라의 영상 스트리밍 데이터의 중요도가 떨어질 경우에 사용하는 자원 분배 기법이다. 그리고 다중 스트림 스케줄링은 고객이 많아 필요한 카메라의 영상 스트리밍 데이터가 증가하여 적정대역폭을 초과하였을 경우의 자원 분배 기법이다.

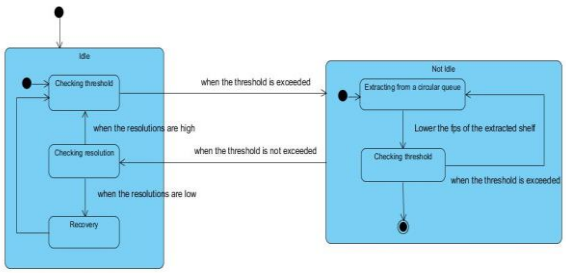
3.1 모션 탐지 기반 대역폭 할당

무인 마켓에 고객이 없을 경우에는 카메라의 영상 스트리밍 데이터의 중요도가 낮아진다. 하지만 기존의 방식은 고객이 없을 때에도 이미지 프로세스를 위한 해상도로 스트리밍하여 굉장한 낭비를 초래한다.

본 논문에서는 대역폭 낭비를 최소화하기 위해 모션 탐지 기반 대역폭 할당을 제안하여 그림 3 과 같이 수행한다. 고객이 없는 일반적인 상황에서 카메라들은 모션을 인식할 수 있는 최소의 해상도로 엣지 디바이스에 스트리밍 데이터를 보내 대역폭 낭비를 줄인다. 하지만 고객이 상품을 담기 위해 매대에 다가서면 엣지 디바이스에서 스트리밍 데이터를 처리하여 고객의 모션을 인식한다. 그 후 이미지 프로세스를 위한 해상도로 전환하여 엣지 디바이스로 스트리밍 데이터를 보내 정상적인 상품 선택 서비스를 제공한다. 상품 선택이 완료된 후에는 다시 최소의 해상도로 전환함으로써 대역폭 낭비를 줄인다.



(그림 3) 모션 탐지 기반 대역폭 할당의 시퀀스 다이어그램



(그림 4) 다중 스트림 스케줄링의 상태 다이어그램

3.2 다중 스트림 스케줄링

터미널 디바이스에서 엣지 디바이스로 데이터를 송신할 수 있는 네트워크 대역폭은 한정되어있다. 그렇기 때문에 수 많은 터미널 디바이스로부터 받는 데이터가 대역폭 임계점을 넘는 경우가 생길 수 있다. 대역폭 임계점을 넘게 되면 패킷 지연시간과 손실률 증가를 초라하게 되고, 최악의 경우에는 네트워크 자체가 다운이 될 수 있다.[6]

이와 같은 문제를 막고자 그림 4 와 같이 수행되는 다중 스트림 스케줄링을 제안한다. 이를 적용하기 위해 모션 탐지 기반 대역폭 할당에서 고객이 매대 앞에 접근하여 해상도를 올릴 때, 원형 큐에 해당 매대의 번호를 넣어준다. 큐의 크기가 일정수치를 넘어 총 소모 대역폭이 대역폭 임계점을 넘는지 판단한다. 대역폭 임계점을 넘을 경우 원형 큐에서 우선순위가 낮은 매대부터 차례대로 해당 데이터 매대의 최대 전송 속도를 낮춘다. 그리고 모션 탐지 기반 대역폭 할당을 적용하여 매대에서 고객이 이탈할 경우 해상도를 낮추면서 해당 매대를 큐에서 제거한다. 이 과정에서 총 소모 대역폭이 대역폭 임계점을 넘지 않는다면 남은 매대의 최대 전송 속도를 모두 복구한다. 이를 통해 낭비되는 대역폭을 최소화하여 엣지 디바이스의 대역폭을 효율적으로 사용할 수 있도록 한다.

또 다른 방식으로 해당 매대의 fps(frame per second)를 낮추는 방법이 존재한다. 하지만 실험을 통해 fps 를 낮춰도 해당 매대의 대역폭이 크게 줄어들지 않을 뿐만 아니라 이미지 프로세싱 하는 데에 있어서 크게 효과를 보지 못하는 것을 확인하였다. 따라서 fps 를 낮추는 것이 아니라 최대 전송 속도를 낮춤으로써 대역폭을 조절하였다.

4. 실험

4.1. 실험환경

본 논문에서 제시하는 모션 탐지 기반 대역폭 할당, 다중 스트림 스케줄링 모두 멀티미디어 프레임워크인 FFMPEG 과, RTMP(Real Time Message Protocol)을 사용하여 실시간 스트리밍을 구현하였고, H.264 를 비디오 코덱으로 사용하였다. 또한 오디오는 사용할 필요가 없으므로 제거하였으며, 최대 전송 속도를 400K/200K 로, 버퍼 크기는 200K/100K, 프레임 속도는 25fps, 가변 비트레이트 화질우선 인코딩(crf)은 23 방식으로 스트리밍 하였다.

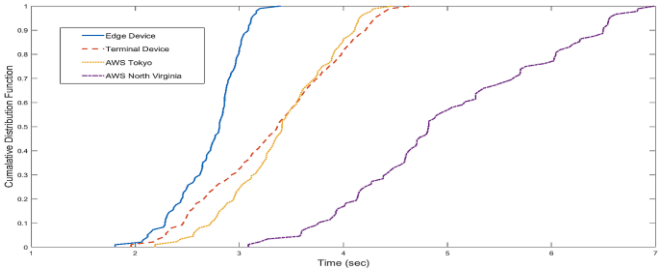
모션 탐지 기반 대역폭 할당을 실험하기 위해 고객이 매장에 들어오지 않았을 경우 모션을 인식할 수 있는 제일 낮은 해상도인 160*90 해상도로 스트리밍 한다. 그리고 고객이 들어왔을 때 해당 카메라는 해당 이미지 프로세스에 필요한 해상도(QR 코드: 320*180, 얼굴 인식: 960*540, 물체 감지: 800*450)로 높여 스트리밍 한다. 마트의 상황은 20 분 동안 20 명의 고객이 들어왔다가 나가는 상황을 가정하고 실험한다.

다중 스트림 스케줄링을 실험하기 위해 매장 내에 있는 5 개의 매대 카메라에서 엣지 디바이스로 영상을 송신하는 것을 가정한다. 그리고 성능 비교를 위해 본 논문에서 제안한 기법을 적용했을 때와 적용하지 않았을 때를 비교하여 소모 대역폭을 기준으로 측정한다. 각 매대의 카메라는 실험의 편의를 위해 가장 높은 해상도를 필요로 하는 얼굴 인식 카메라 한 대를 사용하여 실험한다. 첫 번째 실험과 마찬가지로 고객이 매대 앞에 오지 않으면 모션을 인식할 수 있는 가장 낮은 해상도로 스트리밍 한다. 그리고 고객이 앞에 올 경우 앞서 설명한 얼굴 인식 이미지 프로세스에 필요한 조건으로 스트리밍 한다. 5 곳 모두 고객이 오면 매대의 우선순위가 낮은 순으로 공평하게 5 초간 스케줄링 하여 얼굴인식을 할 수 있는 최소 조건인 최대 전송 속도를 200k 로 낮춘다. 마트의 상황은 500 분 동안 매장 내에 고객이 들어왔다가 물건을 구매하고 나가는 상황을 가정하고 실험한다.

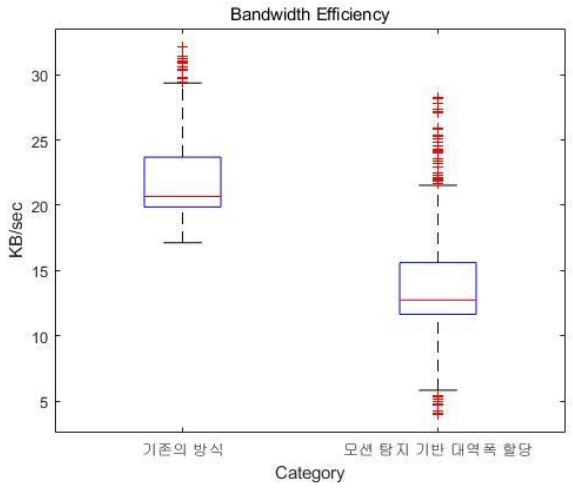
4.2. 실험결과

본 논문에서는 엣지 컴퓨팅의 효율성을 보이기 위해 응답시간을 비교항목으로 설정한다. 그리고 제안한 방법들을 비교 분석하기 위해 단일 매대 소모 대역폭, 총 소모 대역폭을 비교 항목으로 설정한다.

그림 5 는 엣지, 터미널, 클라우드 컴퓨팅(도쿄, 버지니아) 환경에서 매장에 입장한 고객이 물건을 집은 순간부터 사용자 어플까지 정보가 제공되는 응답시간을 CDF(Cumulative Distribution Function)으로 나타낸 그림이다. 충분한 프로세싱 능력을 가지고 있는 도쿄, 버지니아 클라우드 컴퓨팅의 경우 거리가 먼 버지니아의 응답시간이 길어졌고 터미널 컴퓨팅은 거리가 가까워도 프로세싱 능력이 부족하여 응답시간이 길어졌다. 따라서 충분한 프로세싱 능력을 가지고 있



(그림 5) 엣지 컴퓨팅과 클라우드 컴퓨팅 응답시간 비교 결과

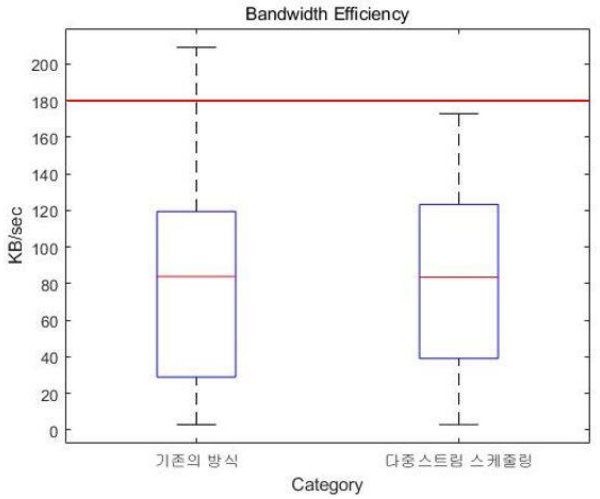


(그림 6) 모션 탐지 기반 대역폭 할당의 대역폭 효율성 비교 박스플롯: 박스 중간의 빨간 선은 중간 값을 의미한다.

고 거리도 가까운 엣지 컴퓨팅 환경이 전체적인 응답시간을 단축시킬 수 있을 뿐만 아니라, 응답시간의 오차율을 대폭 감소시키게 되어 사용자가 불편함을 느낄 확률을 감소시키는 효과가 있음을 보여준다.

그림 6 은 단일 매대 상황에서 모션 탐지 기반 대역폭 할당을 적용하여 대역폭 효율성을 비교한 박스플롯이다. 그림에서 볼 수 있듯이 기존 방식 박스 플롯의 중간 값은 21KB/sec 이고, 모션 탐지 기반 대역폭 할당을 적용한 박스 플롯의 중간 값은 12KB/sec 로 훨씬 더 낮아 좋은 성능을 보인다. 또한 전체적으로도 모션 탐지 기반 대역폭 할당이 대역대역 사용할 때 효율성이 좋다는 것을 알 수 있다.

그림 7 은 기존의 방법과 다중 스트림 스케줄링을 비교한 박스 플롯이다. 각 방식은 매장 내 5 개의 비디오 스트림의 전체 소모 대역폭 데이터를 나타내고 있다. 기존의 방법은 210KB/sec 정도 대역폭을 할당시켜야지만 시스템을 원활하게 사용할 수가 있다. 하지만 다중 스트림 스케줄링을 적용하여 대역폭 임계점을 180KB/sec 로 둘 경우 최대 소모 대역폭이



(그림 7) 다중 스트림 스케줄링의 대역폭 효율성 비교 박스플롯: 윗 부분의 빨간선은 대역폭 임계점을 의미한다.

180KB/sec 를 넘지 않는다. 결과적으로 다중 스트림 스케줄링을 사용하여 적절한 대역폭 임계점을 설정하여 약 14%의 소모대역폭을 줄일 수 있다. 즉, 기존 수용가능 매대 카메라 수에 1.14 배에 해당하는 매대의 카메라를 수용할 수 있다.

이처럼 제안한 기법을 사용하면 총 사용 대역폭을 감소시켜 자원을 보다 효율적으로 사용하여 다중송신 이득이 향상되고 네트워크를 더욱 안정시킬 수 있다.

5. 결론

본 논문에서는 엣지 컴퓨팅의 효율성을 확인하기 위해 Amazon Go 와 같은 무인마켓을 사례연구로 제시하였다. 또한, 엣지 컴퓨팅 내에서의 자원 분배의 효율성 높이기 위해 모션 탐지 기반 대역폭 할당과 다중 스트림 스케줄링을 제안하고 이를 기존의 방법과 비교 분석하였다.

실험결과에서 알 수 있듯이 클라우드, 터미널 컴퓨팅환경보다 엣지 컴퓨팅환경에서의 무인 마켓이 전체적인 응답속도 측면에서 좋을 뿐만 아니라 오차율까지 줄일 수 있다는 것을 증명했다. 이를 통하여 엣지 컴퓨팅의 가장 큰 장점 중 하나인 지연시간 감소를 확인할 수 있었다. 또한 모션 탐지 기반 대역폭 할당을 통하여 대역폭을 효율적으로 사용함으로써 대역폭 낭비를 줄일 수 있었고 다중 스트림 스케줄링을 통하여 대역폭 임계점을 넘지 않도록 설계함으로써 다중송신이득 향상과 네트워크의 안정화를 확인할 수 있었다.

참고문헌

[1] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, Lanyu Xu, “Edge Computing: Vision and Challenges”, IEEE Internet of Things Journal, Volume: 3, Issue: 5, 2016.10.

[2] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, “Mobile-edge computing: Partial computation offloading using dynamic voltage scaling”, IEEE Transactions on Communications, vol. 64, pp. 4268 – 4282, 2016.10

[3] Mahadev Satyanarayanan, "The Emergence of Edge Computing", IEEE Computer, Volume: 50, Issue: 1, 2017.01

[4] T. Q. Dinh, J. Tang, Q. D. La, Q. S. Quek, “Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling.”, Published in the IEEE Transactions on Communications, Vol.65, Issue.8, pp.3571-3584, 2017.08

[5] Amazon Web Service Inc, AWS IoT Greengrass - Developer Guide, 2019.01

[6] 구명모, 정상운, 김상복, “화상회의 시스템을 위한 대역폭 관리 알고리즘 설계 및 구현”, Journal of Korea Multimedia Society, Volume 3, Issue 4, pp. 399-406, 2000.08