

분산 클라우드 환경에서 에지 노드 간의 효율적인 애플리케이션 오프로딩 기법 연구

강석원⁰, 홍충선*

경희대학교 컴퓨터공학과

{dudtntdud, cshong}@khu.ac.kr

Efficient Application Offloading between Edge nodes in Distributed Cloud Environment

SeokWon Kang⁰, ChoongSeon Hong*

Department of Computer Science and Engineering, Kyung Hee University

요 약

최근 클라우드 환경은 중앙 집중형 방식의 클라우드에서 사용자와 가까운 위치에 클라우드 서버를 배치하는 분산형 방식으로 변하고 있다. 이에 따라 수 많은 네트워크 장비를 관리하기 위한 SDN 컨트롤러와 함께 오픈 플로우 스위치로 구성된 에지 노드들을 활용하여 각 노드의 상황에 따라 알맞게 오프로딩할 노드들을 설정하여 부하를 분산 할 수 있다. 본 논문에서는 사용자가 요청한 서비스를 빠른 시간 내에 제공하기 위해 사용자가 요청한 애플리케이션을 가장 적절한 에지 노드에 배치하는 알고리즘을 제안한다. 이는 특정 노드의 사용자 단말기의 트래픽 부하를 줄이고 더욱 빠른 시간 내에 서비스를 제공할 수 있다.

1. 서 론

IoT 관련 서비스의 요구사항이 점차 다양해지고 모바일 사용자의 수가 매년 급증함에 따라, 새로운 형태의 모바일 애플리케이션이 지속적으로 개발되고 있다. 그 중, 얼굴 인식, 자연어 처리, AR, VR과 같은 콘텐츠들을 활용하는 애플리케이션의 경우 상당량의 연산 작업이 필요하며, 지연에 민감하다. 기존에 활용되던 애플리케이션의 경우 많은 연산 작업을 필요로 하게 될 때, 모바일 클라우드 컴퓨팅(Mobile Cloud Computing) 기술을 기반으로 네트워크 연결을 통해 강력한 연산 기능과 저장공간을 활용하였다. 그러나 모바일 디바이스와 모바일 클라우드 간의 거리가 멀 경우에 발생하는 전송 지연 문제와 코어 네트워크와 백홀망으로 트래픽이 집중됨으로써 생기는 병목현상은 저지연을 요구하는 애플리케이션에 치명적으로 적용될 수 있다. 따라서 이러한 문제를 해결하기 위해 사용자와 가까운 거리에 서버를 구성하는 모바일 에지 컴퓨팅(Mobile Edge Computing) 패러다임이 제시되었다[1]. 다양한 지역의 디바이스 연결을 위해 여러 개의 에지 클라우드 서버가 사용자와 가까운 지역에 분산형 방식으로 배치되며, 이는 수 많은 네트워크 장비를 관리하기 위한 Software Defined Network(SDN)과 함께 활용 될 수 있다[2].

SDN 컨트롤러를 통해 에지 클라우드 네트워크 내의 여러 인접한 노드들이 서로 상황을 공유하고 필요에

따라 오프로딩을 통해 특정 에지 노드의 부하를 방지하고, 코어와 백홀망으로 향하는 트래픽 부하를 에지 클라우드 서버로 분산하여, 모바일 디바이스와 서버 간의 빠른 응답을 통해 지연 문제를 해결 할 수 있다.

2. 관련 연구

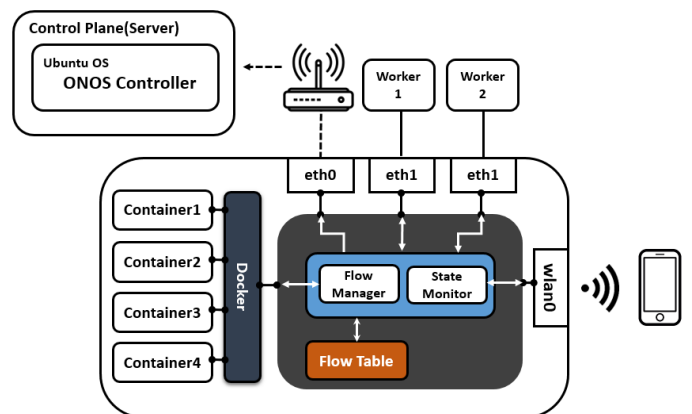


그림 1 Open vSwitch 를 활용한 에지 노드 구성도

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학 ICT 연구센터육성 지원사업의 연구결과로 수행되었음(IITP-2018-2013-1-00717), 아울러 본 연구는 한국정보화진흥원(NIA)의 미래네트워크 선도시험망(KOREN) 사업 지원과제의 연구결과로 수행되었음 (18-951-00-001). *Dr. CS Hong is the corresponding author.

그림 1[3]은 Wi-Fi access point 역할을 할 수 있도록 구축된 에지 노드의 구성도이며, 노드 자체에서 플로우를 제어할 수 있도록 Open vSwitch를 고려하여 구성하였다. 에지 노드에서 Open vSwitch를 활용함으로써 사용자가 요청하는 서비스에 대한 요구사항을 제공할 수 있도록 알맞게 플로우를 설정하여 패킷을 포워딩 할 수 있다. 또한 도커 컨테이너를 활용하여 에지 노드간의 플로우 설정에 필요한 모듈 또는 에지 노드에서 제공될 수 있는 서비스 등을 배치할 수 있다. 최근 IoT 관련 서비스의 요구사항이 다양해지고 있음에 따라 클라우드 환경도 점차 분산화되고 있으며, 에지 노드의 경량화를 통해서 사용자와 가장 가까운 거리에서 서비스를 제공하는 방안을 모색하고 있다[4]. 이러한 요구사항에 맞춰서 본 논문에서는 사용자와 가까운 지역에 여러 개를 분산 배치하기 위한 경량화된 노드 간의 애플리케이션 오프로딩 알고리즘을 제안함으로써 에지 노드를 효율적으로 활용될 수 있도록 하고자 한다.

3. 제안사항

에지 네트워크는 그림 2과 같이 구성되어 있다. 각 에지 노드인 EN_i은 가상 노드(Virtual Nodes)를 활용한 MEC 서버와 Open vSwitch를 활용한 스위치의 역할을 동시에 한다. 또한 각 에지 노드는 SDN 컨트롤러에 의해서 주기적으로 각 노드의 큐, 네트워크 상황을 공유한다.

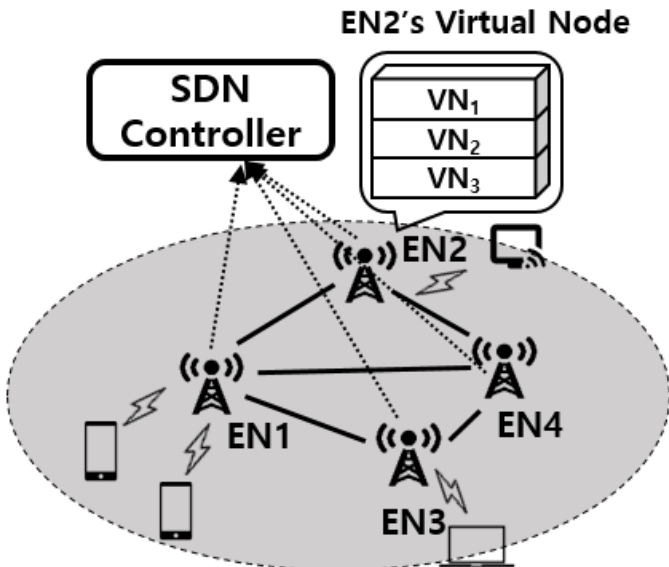


그림 2 모바일 에지 네트워크 모델

본 논문은 각 에지 노드의 애플리케이션 실행 큐, 트래픽 상황, 프로세싱 타임을 고려하여 각 노드에 가중치를 설정하고, 네트워크 상황을 고려하여 가장 적절한 에지 노드에 실행될 애플리케이션을 배치한다. 노드에 대한 가중치의 공식은 다음과 같다.

$$W_i = \frac{\sum_{k=1}^n C_k}{P_i} + T_i^{send} + T_i^{rcv}$$

표 1. 노드 선택 표기법

Symbols	Meaning
EN _i	서비스 노드 i(= V _i)
e _j	j번째의 간선
W _i	EN _i 의 노드 가중치
Q _i	EN _i 큐 내의 총 태스크 크기(Mbyte)
T _i ^{send}	출발 노드에서 EN _i 까지의 전송 속도(MB/s)
T _i ^{rcv}	EN _i 에서 도착 노드까지의 전송 속도(MB/s)
C _k	에지 노드 큐에서 k번째 태스크 크기(Mbyte)
P _i	EN _i 에서의 애플리케이션 처리 속도(CPU 성능)

그림 2는 에지 노드와 그에 해당하는 태스크 큐를 임의로 설정하고, 가중치를 두어 가장 알맞은 경로로 라우팅 하는 예시이다. 이때, 모든 노드는 유저가 요청한 서비스를 제공 할 수 있다고 가정한다.

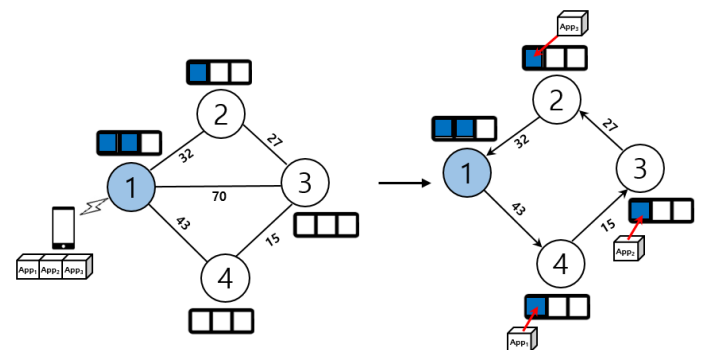


그림 3 에지 노드 구성도

실행해야 하는 각 애플리케이션마다 노드의 W_i의 가중치를 적용하고, 가중치가 가장 낮은 에지 클라우드에 해당 애플리케이션을 할당하여 실행케 한다. 또한 각 에지 노드의 큐의 상황과 네트워크 상황을 고려하여 Shortest Path 알고리즘을 통해 최선의 경로를 선택한다.

Algorithm 1. Edge Node Selection for Offloading

```

Input :  $C_k, P_i, Q_i$ 
Output : NodeList[] for optimal Offloading.
1: Initialize NodeList = [],
    $T_i^{send} = \text{shortest\_path\_cost}(EN_1, EN_i),$ 
    $T_i^{rcv} = \text{shortest\_path\_cost}(EN_i, EN_1)$ 
2: for  $i = 1$  to  $k$ 
3:    $Q_i += C_i$ 
4: for all Tasks do
5:   for all  $EN_i$  do
6:      $W_i = (Q_i / P_i) + T_i^{send} + T_i^{rcv}$ 
7:     NodeList  $\leftarrow \min\{W_1, W_2, \dots, W_i\}$ 
8:      $Q_i += \text{Task}$ 
9:     If(NodeList == empty) :
10:       $T_i^{send} = \text{shortest\_path\_cost}(EN_1, EN_i)$ 
11:       $T_i^{rcv} = \text{shortest\_path\_cost}(EN_i, EN_1)$ 
11:     Else :
12:       $T_i^{send} = \text{shortest\_path\_cost}(\text{NodeList}[i-1], EN_i)$ 
13:       $T_i^{rcv} = \text{shortest\_path\_cost}(\text{NodeList}[i-1], EN_i)$ 
    
```

4. 성능평가

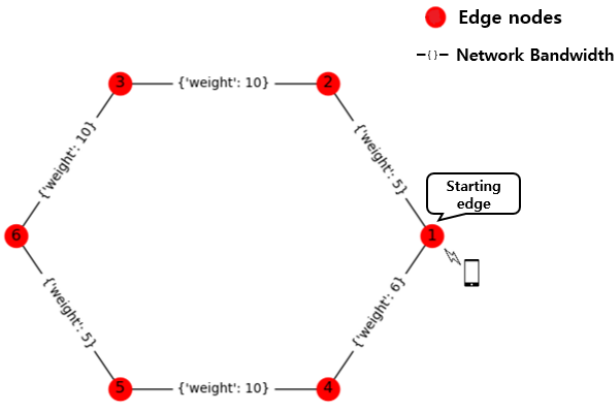


그림 4 성능 평가를 위한 그래프

EN의 개수가 총 6개인 그래프를 사용하여 성능평가를 진행하였다. 이때, e_j 의 가중치 값은 해당 노드 사이의 데이터 전송 속도이며, 1~12.5 MB/s중 임의의 값을 사용하였다. 각 노드는 고유값으로 태스크 큐의 값인 C_k 와 프로세스 성능인 P_i 를 가지고 있다. 이 세개의 인자들을 활용하여 노드 가중치인 W_i 를 계산하고 낮은 순서대로 사용자가 요청한 애플리케이션을 오프로딩 한 후에 최단 경로로 라우팅하여 결과값을 받도록 한다. 사용자가 연결된 에지 클라우드 서버는 1번이라고 가정하며, 각 애플리케이션의 크기는 동등하다. 본 논문에서는 기존의 방식인 모든 애플리케이션을 1번 MEC 서버에 요청하는

방식과 오프로딩 할 노드를 임의적으로 선정하는 방법, 그리고 알고리즘을 활용하는 방법으로 서비스 요청 반환 시간을 비교하였다.

그림 4는 노드가 6개인 그래프에서 알고리즘을 사용하였을 때, 모든 애플리케이션을 단말기와 직접 연결된 MEC 서버에서 실행하였을 때, 그리고 임의의 노드로 오프로딩 하였을 때 결과이다. 기존의 한 에지 노드에서 사용자의 요청을 대응할 때보다 약 30% 정도의 성능 향상을 보인다.

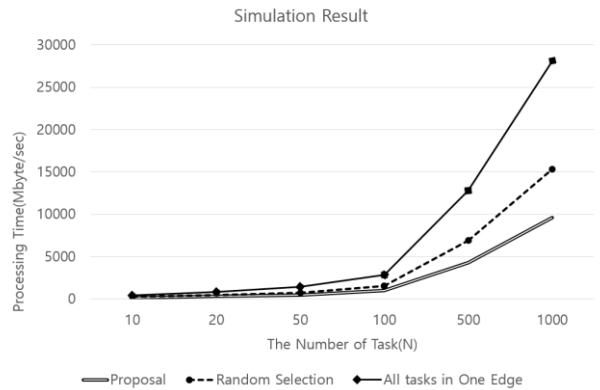


그림 5 시뮬레이션 결과 그래프

5. 결론

본 논문에서는 Open vSwitch를 통하여 구성된 에지 클라우드 환경에서 모바일 디바이스와 단일 에지 노드의 한계를 극복하기 위해 다수의 에지 노드를 활용한 오프로딩 알고리즘을 제시하였으며, 기존 단일 에지 노드의 오프로딩 방법과 비교하여 더욱 높은 효율을 보였다. 알고리즘을 개선시키기 위해 실제 경량화된 에지 노드에 시스템을 적용하고, 다양한 변수들을 고려할 필요가 있기 때문에 이에 대한 연구를 진행할 예정이다.

6. 참고문헌

- [1] 김상기, 박종대, "5G를 위한 MEC 기술동향", 전자통신동향 분석, 제31권 1호, PP 25~35, 2016년 2월.
- [2] Chaitanya Aggarwal, Kingshuk Strivastava, "Securing IoT devices using SDN and edge computing", 2016 2nd International Conference on Next Generation Computing Technologies(NGCT)
- [3] 강석원, 홍충선, "분산 클라우드 환경에서 효율적인 패킷 전송을 위한 오픈플로우 스위치 기반의 에지 클라우드 구조 연구", 2018년 한국컴퓨터종합학술 대회 논문집
- [4] Adisorn Lertsinsrubeetavee, Anwaar Ali, Carlos Molina-Jimenez, Arjuna Sathiaselan, Jon crowcroft, "PiCasso: A Lightweight Edge Computing Platform," 2017 IEEE 6th international Conference on Cloud Networking(CloudNet), October. 2017.