

분산 클라우드 환경에서 에지 클라우드 활용을 위한 서비스 인지 라우팅 모듈 구현

김도현[○], 김윤곤, 변준영, 손동영, Tien-Dung Nguyen, 홍충선*
 경희대학교 컴퓨터공학과
 { doma[○], ykkim, bji557, dudehd1123, ntiendung, cshong* }@khu.ac.kr

Service-Aware Routing Module for Edge Cloud in Distributed Cloud Environment

Do Hyeon Kim[○], Yunkon Kim, Jun-Young Byun, Dong-Young Son, Tien-Dung Nguyen, Choong Seon Hong*

Department of Computer Science and Engineering, Kyung Hee University

요 약

클라우드 컴퓨팅 환경은 네트워크 서비스 수요의 증가와 더불어 다양한 형태의 서비스가 생겨남에 따라 실시간, 저 지연 등의 요구사항을 만족하기 위해 점차 중앙 집중형 방식에서 분산화 방식으로 변화하고 있다. 분산화 방식의 경우, 사용자와 지리적으로 멀리 떨어져 있는 클라우드 환경을 Core 클라우드, 사용자에게 보다 가까이 위치해 있는 클라우드를 에지 클라우드로 구분할 수 있다. 본 논문에서는 사용자 가까에서 실시간, 저 지연등의 높은 네트워크 성능을 요구하는 서비스를 제공하기 위한 에지 클라우드 프로토타입을 구축하고, 서비스별로 패킷 전송경로를 제어할 수 있는 라우팅 모듈을 제안한다. 이는 기존의 코어 클라우드에서 서비스를 받을 때 보다 높은 네트워크 성능을 기반으로 서비스를 제공받을 수 있도록 하는 역할을 하며, 실제 코어 클라우드 대비 에지 클라우드에서의 최대 대역폭 사용량이 높게 측정 되었다.

1. 서 론

최근 사용자의 서비스 품질 요구사항이 지속적으로 증가하고, 다양한 형태의 서비스가 개발됨에 따라 클라우드 및 네트워크 분야에서 실시간, 저 지연 등의 요구를 만족시키기 위한 다양한 관점에서의 연구가 진행되고 있다 [1-2]. 특히, AR(Augmented Reality)과 VR(Virtual Reality) 서비스의 등장은 최적의 서비스 제공을 위한 컴퓨팅 및 네트워킹 연구의 필요성을 더욱 부각시키고 있다. 이를 위해, 최근 클라우드 환경은 중앙관리의 환경에서 분산 관리 환경으로 분리되고 있으며, Fog/Edge 컴퓨팅, Mobile Edge 컴퓨팅, 마이크로 데이터센터와 같은 다양한 형태의 분산 클라우드 기술들이 제안되었다. 이와 같은 분산형 클라우드 환경을 구축하고 실현하기 위해, 클라우드와 네트워크 중점의 연구가 진행되어야 할 것이다 [3]. 본 논문에서는 분산 클라우드 환경에서 활용될 Edge 클라우드의 프로토타입을 구축하고, 이를 활용하기 위한 라우팅 모듈을 제안한다. 프로토타입의 경우, Wi-Fi 모듈을 기반으로 Access Point 기능을 갖추으로써 모바일 장비가 연결될 수 있도록 하였고, 라우팅 모듈의 경우 NAT(Network Address Translation) 기능을 기반으로 개발하였다. 본 논문의 2장에서는 분산 클라우드의 개념 및 관련 연구들의 동향을 살펴보고, 3장에서는 논문에서 제안하고 있는 Edge 클라우드의 구조 및 라우팅 모듈에 대해 설명한다.

성능평가의 경우 4장에서 iperf를 통한 Core 및 Edge 클라우드 활용시의 성능비교와 Delay에 대한 내용을 다루며 5장에서 본 논문을 요약·정리하고 향후 연구 계획을 제시한다.

2. 관련 연구

2.1 분산 클라우드

Amazon AWS, Google Cloud, Microsoft Azure와 같은 기존 클라우드는 서비스 사용자로 하여금 원하는 만큼 서비스를 이용하고 비용을 지불하여 실제 인프라를 구축하는 비용을 절감 할 수 있다. 하지만 네트워크의 관점에서 봤을 때, 이러한 클라우드 서비스는 사용자와의 거리가 멀리 떨어져있다는 단점이 있기 때문에, 지연에 민감한 서비스를 제공하고자 할 때 문제가 발생하게 된다[3].

이에 본 논문에서는 사용자가 네트워크를 사용함에 있어 가장 첫 번째로 접근하게 되는 노드인 네트워크 Edge 노드를 클라우드화 시킴으로써 사용자 가까에서 지연을 최소화 함과 동시에 서비스를 제공해줄 수 있는 노드로 정의하고 이를 Edge 클라우드라고 명시한다. 기존 네트워크에서 Base Station(BS), Access Point(AP), 스위치 등이 Edge 노드로 포함될 수 있으며 본 논문에서는 Wi-Fi 네트워크 사용을 위한 Desktop 기반의 AP를 구축하여 이를 Edge 클라우드로 활용하였다.

2.2 Mobile Edge Networks

[4]에서는 Edge 노드에서 활용될 수 있는 세 가지 서로 다른 기술을 정리하였다. 먼저 Mobile Edge Computing(MEC) 기술의 경우, MEC 서버를 기반으로 BS 가까이 위치하여 사용자의 서비스요청에 대한 응답을 직접 해주거나 데이터 센터와

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학 ICT 연구센터육성 지원사업의 연구결과로 수행되었음 (IITP-2017-2013-0-00717)
 *Dr. CS Hong is the corresponding author

연계하여 해당 요청에 대응하는 기술이다. Fog Computing은 IoT를 위한 플랫폼으로써 디자인 되었으며, 주요 특징은 사용자 인접 장치 간의 연동을 통해 Task를 해결할 수 있다는 점이다. 즉, BS 또는 차량, AP 등이 Fog 노드가 될 수 있으며 모바일 기기는 요구 사항에 가장 알맞은 Fog 노드를 선택하여 서비스를 요청하거나 전달받게 된다. Cloudlet의 경우 본 논문에서 제안하는 Edge 클라우드와 가장 비슷한 개념으로써, Wi-Fi Access Point 또는 LTE BS에 배치될 수 있다. 이는 저 지연을 요구하는 서비스를 효율적으로 제공하기 위해 제안되었으며 클라우드 관점에서 디바이스-Cloudlet-클라우드의 3계층 구조를 보여주고 있다. 본 논문에서는 클라우드 기술과 더불어 네트워크 관점에서의 Access 계층을 적극 활용함으로써 기존 클라우드의 보조역할을 할 수 있는 Edge 클라우드의 프로토타입을 실제 구축하고 Core 클라우드 대비 네트워크 성능 측면에서의 이점을 확인하고자 한다.

3. 제안사항

3.1 Edge 클라우드

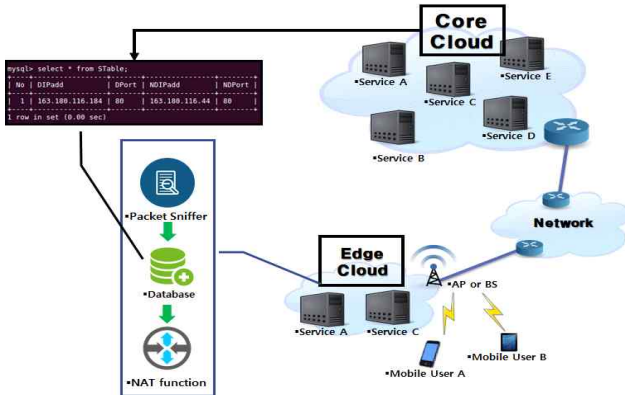


그림 1 분산 클라우드 네트워크 토폴로지

그림 1은 분산 클라우드를 활용하는 네트워크 토폴로지로, Amazon AWS 와 같은 Core 클라우드가 있으며 사용자 가까이에 위치해 있는 AP 또는 BS를 클라우드화하여 Edge 클라우드로 활용하는 모습을 나타낸 그림이다. Edge 클라우드 내에는 Packet Sniffer, 서비스 매칭을 위한 Database와 최종 라우팅을 결정하는데 사용되는 NAT 기능이 있으며 Edge 클라우드에서 제공되어야 할 서비스가 사용자로부터 요청된다면, NAT 기능을 통해 패킷의 목적지 주소를 변환하여 Edge 클라우드로부터 제공 받을 수 있도록 한다.

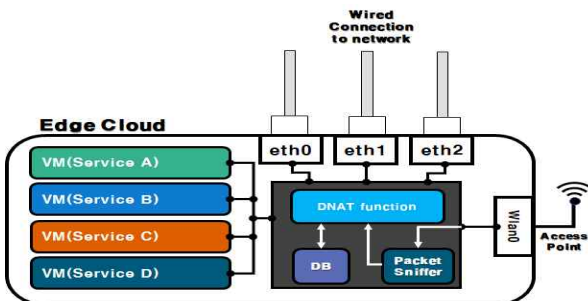


그림 2 Edge 클라우드 구조

그림 2는 Edge 클라우드의 구조를 나타낸 그림으로써, Ubuntu 14.04 기반의 Desktop 환경을 이용하였으며, 무선 네트워크를 활용하기 위해 Atheros AR9485 모듈을 탑재하고 AP 환경을 구성하여 모바일 기기들이 Edge 클라우드의 AP를 통해 네트워크에 접속할 수 있도록 하였다. 또한 여러 개의 VM을 두어 서비스를 올릴 수 있도록 함으로써, DNAT 기능과의 연계를 통해 실시간, 저 지연을 보장해야 하는 서비스를 제공할 수 있도록 설계하였다.

Packet Sniffer 기능의 경우, 서비스를 인지하기 위한 초기 단계로써 패킷을 분석하는 모니터링 모듈이며 Edge 클라우드에서 제공되어야 할 서비스가 사용자로부터 요청되면 NAT 기능을 활용하여 라우팅을 결정하도록 전달해주는 역할을 한다. Database의 경우, Core 클라우드에서 특정서비스를 Edge 클라우드에서 제공해야 할 서비스로 등록하기 위해 활용된다. 하지만 본 논문에서는 어떤 서비스가 Database로 등록되는지에 대해서는 다루지 않는다.

3.2 Service-Aware 라우팅 모듈

알고리즘 Service Aware Routing Algorithm

```

1: DIPadd ← original destination IP address from DB
2: DPort ← original destination Port number from DB
3: NDIPadd ← new destination IP address from DB
4: Loop packet monitoring
5: if packet has DIPadd and DPort then
6:   check the iptables of PREROUTING
7:   if DIPadd in matchField of iptables then
8:     print Packet Data with payload information
9:   else
10:    chain = iptc.Chain(iptc.Table( 'nat' ), 'PREROUTING' )
11:    rule = iptc.Rule()
12:    rule.protocol = 'tcp'
13:    rule.dst = DIPadd
14:    m = rule.create_match( 'tcp' )
15:    m.dport = DPort
16:    t = rule.create_target( 'DNAT' )
17:    t.to_destination = NDIPadd
18:    chain.insert_rule(rule)
19:    print 'Routing rule is updated'
20:   else
21:    print Packet Data with payload information
    
```

Service Aware(SA) 라우팅 모듈은 Python 언어로 개발하였으며, 그림 2의 Packet Sniffer 기능과 DNAT 기능을 통해 동작한다. Packet Sniffer 기능의 경우, 평상시에는 모니터링 모듈로써 활용되며 서비스가 인지될 경우에 DNAT 기능과 연동하여 라우팅을 결정하게 된다. 위와 같은 알고리즘을 통해 서비스를 인지할 수 있으며, DNAT 기능을 통한 목적지 주소 변환을 기반으로 라우팅이 이루어진다. 이때, NDIPadd 같은 경우 Edge 클라우드 내의 VM의 주소가 맵핑됨에 따라 Edge 클라우드로부터의 서비스 제공이 가능하다. 아울러 DNAT 기능의 경우 Python iptc 모듈을 활용하여 구현하였으며, PREROUTING을 수행함으로써 매칭되는 패킷들을 새로운 주소로 전송한다[5].

4. 성능 평가

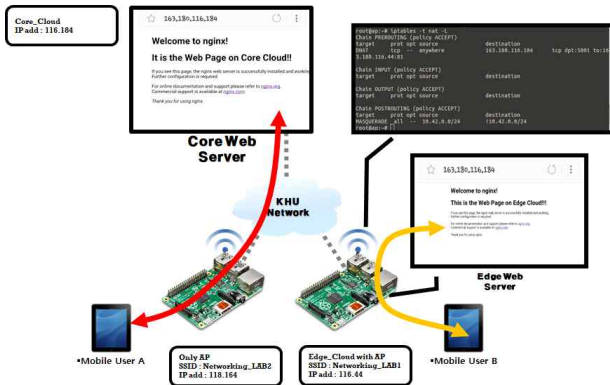


그림 3 라우팅 모듈 동작 테스트 결과

그림 3은 라우팅 모듈 테스트 결과를 나타낸 그림으로, Mobile User A가 116.184의 IP address를 Chrome에 입력한 결과와 Mobile User B가 같은 IP address를 입력한 결과를 나타낸 것이다. 본 테스트를 시작하기 전에 Database상에 116.184로 요청되는 서비스와 관련된 패킷들을 116.44로 맵핑할 수 있도록 정보를 등록하였다. 주소창에 116.184를 입력하는 경우, Mobile User A의 결과와 같이 웹 페이지에 <It is the Web Page on Core Cloud>라는 문장이 출력되어야 하지만 User B의 경우는 Edge 클라우드에서 라우팅 모듈이 Core로 가는 패킷을 Edge로 전달하기 때문에 <It is the Web Page on Edge Cloud>라는 문장이 출력되고 있다. 또한 iptables -t nat -L 명령어를 통해 DNAT 테이블을 확인한 결과, 116.184로 가는 패킷들을 새로운 목적지 주소인 116.44로 변경될 수 있도록 룰이 세팅되어 있다.

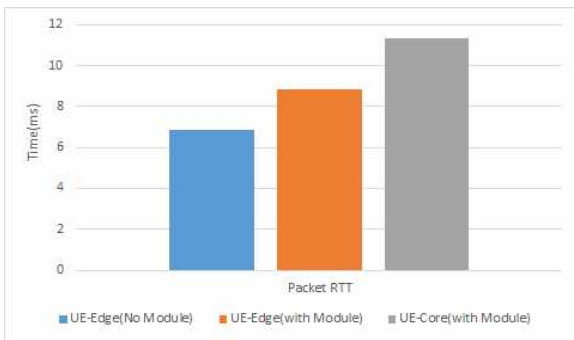


그림 4 모듈 딜레이 테스트 결과

그림 4는 모듈 사용에 대한 딜레이를 테스트한 결과이다. UE-Edge(with Module)은 기존에 UE가 Core로 패킷을 보내고자 할 때, 라우팅 모듈에 의해 Edge로 전송되는 경우를 뜻하며 UE-Core(with Module)은 라우팅 모듈은 동작하지만 Core로 패킷을 보내는 경우를 뜻한다. Packet RTT의 경우, 목적지까지의 Round Trip Time (RTT)을 측정된 결과이며 모듈 사용으로 인한 딜레이가 발생하지만 Core 클라우드와 사용자 기기, Edge 클라우드와 사용자 기기 두 RTT를 비교해 보았을 때, 상대적으로 미미한 부분이라 판단된다.

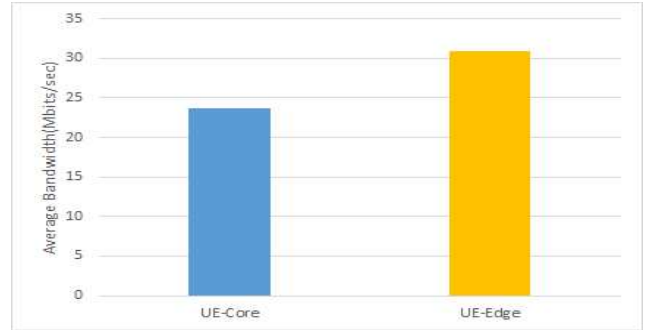


그림 5 평균 대역폭 사용량

그림 5는 평균 대역폭 사용량을 나타낸 결과로써, iperf 테스트 툴을 기반으로 평가를 진행하였다. User Equipment (UE)가 Client 역할을 하고 Core와 Edge가 각각 iperf Server 역할을 하도록 설정하여 진행한 결과, 평균 대역폭 사용량이 Edge 노드를 사용할 때가 약 30.88% 증가하였다. 이는 사용자와 가까운 노드에서 서비스를 제공할 때, 보다 높은 대역폭을 활용할 수 있음을 보여준다.

5. 결론 및 향후 연구

본 논문에서는 분산 클라우드 환경에서 Edge 클라우드를 활용하기 위한 서비스 인지 라우팅 모듈을 제안하였다. DNAT 기능을 활용한 라우팅 모듈을 통해 Edge 클라우드에서 서비스를 제공할 수 있도록 동작 시킬 수 있었으며, 상대적으로 사용자와 멀리 떨어진 Core를 활용할 때 보다 더 높은 대역폭 사용량을 보였다.

향후 연구로는 Core 클라우드가 갖고 있는 서비스들 중 어떠한 서비스를 Database에 등록하고 Edge 클라우드에서 제공할 것인가에 대한 연구를 진행 할 것이며, 모듈 사용에 대한 딜레이를 줄일 수 있는 방안을 연구할 계획이다.

6. 참고 문헌

- [1] Pradip Kumar Sharma, Mu-Yen Chen, Jong Hyuk Park, "A Software Defined Fog Node based Distributed Blockchain Cloud Architecture for IoT," Accepted in IEEE Access, DOI: 10.1109/ACCESS.2017.2757955
- [2] Swaroop Nunna, Apostolos Kousaridas, Mohamed Ibrahim, Markus Dillinger, "Enabling Real-Time Context-Aware Collaboration through 5G and Mobile Edge Computing," 2015 12th International Conference on Information Technology - New Generations, 13-15 Apr, Las Vegas, USA, 2015
- [3] Yunkon Kim, Dong-Young Son, Jun-Young Byun, Eui-Nam Huh, "Study on Cloud Gradual Decentralization Model for Distributed Cloud Environment," 2017 Korea Computer Congress (2017 KCC), 18-20 June, Jeju, Korea, 2017(in Korean)
- [4] Shuo Wang, Xing Zhang, Yan Zhang, Lin Wang, Juwo Yang, Wenbo Wang, "A Survey on Mobile Edge Networks : Convergence of Computing, Caching, and Communications," IEEE Access, Vol.5, 2017
- [5] Github, "Welcome to Python-Iptable's Documentation!" Online Access : <http://dx.github.io/python-iptables/>