SDN 기반 모바일 엣지 컴퓨팅 환경에서 머신러닝을 이용한 태스크 오프로딩 방안 연구

김기태, 홍충선*

경희대학교

glideslope@khu.ac.kr, *cshong@khu.ac.kr

A Study on Task Offloading Method Using Machine Learning in Software Defined Network Based Mobile Edge Computing

Kitae Kim, Choong Seon Hong*

*Kyung Hee University

요 약

통신기술의 발전과 다양한 모바일 기기의 개발 등으로 모바일 트래픽이 급증하는 요즘 혼잡한 네트워크를 효율적으로 관리하고 사용자들이 요구하는 서비스를 보장하기 위한 다양한 기술들이 발전되었으며 그 중 대표적인 것이 소프트웨어 정의 네트워크와 모바일 엣지 컴퓨팅이다. 본 논문에서는 SDN 기반 모바일 엣지 컴퓨팅 환경에서 제한된 자원의 모바일 기기 사용자가 엣지 노드로 태스크 오프로딩을 할 때 네트워크 환경과 엣지 노드의 자원 및 예상 실행시간을 고려한 강화학습 기반 오프로딩 방안을 제시한다.

Ⅰ. 서 론

MEC(Mobile Edge Computing)은 기존의 클라우드와 다르게 기지국이나 AP와 같은 네트워크의 가장자리에 컴퓨팅 자원을 배치시켜 사용자와 조금 더 가까운 위치에서 저 지연, 고 대역폭의 서비스를 제공하는 미래 네트워크의 핵심 기술이다. 이러한 엣지 노드는 사용자에게 미리 캐싱된 컨텐츠나 컴퓨팅 자원을 제공 할 수 있으며 이러한 서비스들은 백홀망을 거치지 않기 때문에 백홀망의 대역폭을 줄일 수 있는 동시에 사용자에게는 저 지연의 서비스를 제공할 수 있는 것이 장점이다[1-2].

최근 떠오르는 빅 데이터(Big Data), 가상현실, 증강현실 기술로 인하여 다양한 어플리케이션이 등장하였으며 이러한 어플리케이션들은 고 성능의 하드웨어 스펙을 요구하기 때문에 현재의 모바일 기기에서의 처리가 힘든 경우가 있으며 배터리 수명 또한 적절하지 않다. 하지만 통신기술의 발전으로 이러한 어플리케이션을 위한 데이터 전송이 가능케 되고 모바일 디바이스에서 모바일 엣지 노드로의 오프로딩 기술로 가능하게 되고 있다. 오프로딩 기술이란 연산에 필요한 데이터만을 엣지 노드에 전송하고 실제 연산은 엣지 노드에서 실행되고 이로부터 결과를 전송받는 기술이다. 따라서 모바일 기기는 배터리를 소모를 줄이고 빠른 어플리케이션 서비스를 이용할 수 있다. 이러한 오프로딩을 위해서는 네트워크 상황과 엣지 노드의 활용 가능한 자원들을 고려해 오프로드 시켜야 하며 해당 어플리케이션이 모바일 기기에서 실행 가능 할 시 오프로드를 시킬지에 대한 여부, 어떠한 노드로 오프로드를 시킬 것인지에 대한 활발한 연구가 진행되고 있다. 본 논문에서는 선형회귀를 통한 태스크의 수행시간, 현재 엣지 노드의 자원상황들을 고려해 강화학습 기법중 하나인 Q-Learning을 이용하여 엣지노드로 태스크를 오프로딩 하는 기법을 제안한다.

Ⅱ. 제안사항

선형회귀를 이용한 태스크 수행시간 예측

CPU_{req}	MEM_{req}	$Disk_{req}$	CPU_{cur}	MEM_{req}	$Disk_{req}$	$Time_{\exp}$
15%	44%	5%	52%	71%	12%	2.2
50%	31%	31%	41%	14%	22%	5.7
80%	23%	52%	10%	25%	48%	4.1

표 1 .태스크 수행시간 예측을 위한 데이터

표1은 각 태스크의 수행시간을 예측하기 위한 데이터 셋의 예시이다. CPU_{req},MEM_{req},$Disk_{req}$는 태스크의 자원 요구량을 나타내며 실제 수행시간과 함께 코드 프로파일링[3]을 통해 얻을 수 있다. CPU_{cur},MEM_{req},$Disk_{req}$는 실제 태스크가 실행 된 노드의 현재 자원상황이며 위와 같은 데이터로 학습된 선형회귀 모델은 실행시간 $Time_{\exp}$를 예측하기 위해 6개의 입력을 필요로 한다.

Q-Learning 기반 태스크 오프로딩

일반적으로 태스크는 오프로딩 가능한 태스크와 오프로딩 불가능한 태스크로 나뉘며 로컬 디바이스의 카메라, 센서 등을 이용하는 메소드의 경우 오프로딩 불가능한 태스크로 분류한다. 위 시스템 모델에서 모바일 유저는 SDN 컨트롤러로 서비스 요청을 보내고 SDN 컨트롤러는 각 태스크가 각 엣지 노드에서의 수행시간을 예측한다. 이러한 예측 값과 네트워크 상태를 고려해 오프로딩 할 것인지 로컬에서 수행할 것인지를 판단할 수

있다.

$$Reward = \frac{1}{Transmission\,Delay + Time_{\exp}} + Bandwidth$$

수식 2. 보상함수

수식 2는 제안하는 시스템에서 Q-Learning을 적용하기 위한 보상함수이다. 노드까지의 Transmission Delay와 예측된 수행시간이 적을수록, 그리고 대역폭이 클수록 큰 보상을 얻게 되며 반대의 경우에는 적은 보상을 받게 된다. 따라서 보상의 합을 최대로 만들기 위하여 행동하는 에이전트는 위와 같은 보상함수에 따라 오프로딩을 할 것인지 하게 되는 경우 어떤 노드로 오프로딩을 할 것인지 결정을 하게 된다.

알고리즘1. Q-Learning Based Task Offloading	
1:	**//Execution Time Prediction**
2:	en_Num = Number of Candidate Edge Node
3:	**S** -> (time[EN], Transmission Delay, Bandwidth)
4:	**A** -> (mobile device)
5:	**Task Request from User**
6:	Req($CPU_{req}, MEM_{req}, Disk_{req}, CPU_{cur}, MEM_{req}, Disk_{req}$)
7:	**for** $EN \in EN_{all}$
8:	time[EN]=prd($CPU_{req}, MEM_{req}, Disk_{req}, CPU_{cur}, MEM_{req}, Disk_{req}$)
9:	**if** time[EN] > $time_{limit}$
10:	time[EN] = -1
11:	**end if**
12:	**end for**
13:	return time[EN]
14:	**// Q-Learning Process**
15:	**while(n epochs)**
16:	Choose Edge Node Randomly with State **S**
17:	Reward = reward_function()
18:	Update Q-Function **Q(S,A)**
19:	**end while**

표 2 . 태스크 오프로딩을 위한 Q-Learning 과정

시뮬레이션

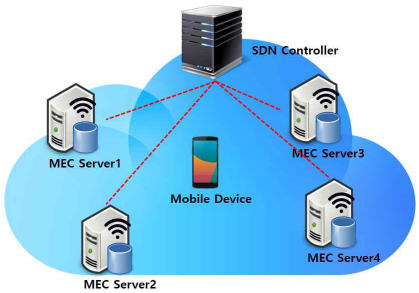

그림 1 . 시뮬레이션 토폴로지

그림2와 시뮬레이션 토폴로지는 MEC노드 4개와 모바일 디바이스 1개, SDN 컨트롤러로 이루어졌으며 MEC 노드들은 지속적으로 SDN 컨트롤러로 자신의 자원 상태 및 채널 상태를 수신한다. 본 논문에서 제안하는 사항을 검증하기 위하여 번호판 인식 프로그램[4]을 이용하였으며 3개의 임의 노드에 고의적인 혼잡상황을 주어 적절한 노드에 오프로딩이 되는지 검증하였으며 이 경우 로컬 디바이스에서의 수행시간과 클라우드 노드에서의 수행시간을 비교 측정하였다.

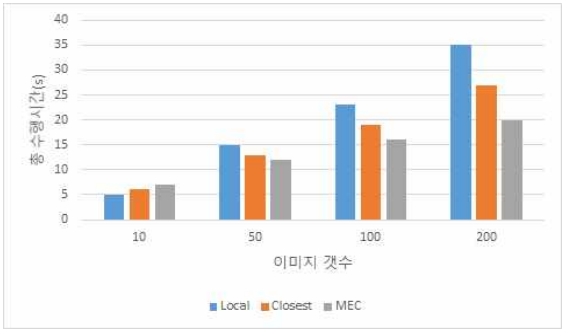

그림 2 . 실험결과

Ⅲ. 결론

본 논문에서는 SDN 기반 MEC 환경에서 모바일 디바이스의 태스크를 효율적으로 오프로드 시키기 위한 기법을 제안하였다. 성능평가 결과 네트워크 혼잡도와 리소스 사용량이 가장 적은 노드로 오프로드 되었으며 오프로드 결정이 되었을 때 로컬 디바이스에서와 가장 가까운 노드, 선택된 노드에서의 총 수행시간을 측정하였을 때 더 적게 걸리는 것을 확인 할 수 있었다. 다만 SDN 컨트롤러에서 머신러닝 및 노드의 상태 데이터를 지속적으로 받기 때문에 규모가 커지게 되는 경우 큰 부하가 발생한다. 이러한 네트워크 규모는 확실한Q-Learning의 성능 검증을 위해 필수적인 사항이므로 이러한 부하를 줄일 수 있는 연구가 필요하며 앞으로 진행 예정이다.

ACKNOWLEDGMENT

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2015-0-00567, 유무선 통합 네트워크에서 접속 방식에 독립적인 차세대 네트워킹 기술 개발). *Dr. CS Hong is the corresponding author

참 고 문 헌

[1]윤찬현 “모바일 엣지 컴퓨팅 기술 동향”, KRnet 2018

[2] Yuyi Mao, Changsheng You, Jun Zhang, Kalbin Huang, Khaled B. Letaief, “A Survey on Mobile Edge Computing The communication Perspective”, IEEE Communications Surveys & Tutorials

[3]프로파일링,https://ko.wikipedia.org/wiki/%ED%94%84%EB%A1%9C%ED%8C%8C%EC%9D%BC%EB%A7%81_(%EC%BB%B4%ED%93%A8%ED%84%B0_%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D)

[4]OpenCV-CarLicensePlateRecognizer,https://github.com/devetude/Open-CV-CarLicensePlateRecognizier/tree/master/car_license_plate_images