강화 학습 기반 무선 충전 모바일 에지 컴퓨팅 네트워크의 지연시간 최적화

오치원, 황상원, 이인규
고려대학교

ohcheewony@korea.ac.kr, tkddnjs3510@korea.ac.kr, inkyu@korea.ac.kr

Deep Reinforcement Learning Approach for Latency Minimization in Wireless Powered Mobile Edge Computing Networks

Cheewon Oh, Sangwon Hwang, Inkyu Lee
Korea Univ.

요 약

본 논문에서는 DDQN(Double Deep Q-Network) 강화학습 방법을 사용하여 단일 사용자 MEC(Mobile Edge Computing) 모델에서 장기적인 관점에서의 지연시간을 최소화하는 문제를 푼다. 또한 시뮬레이션을 통해 제안한 기법의 성능을 확인한다.

Ⅰ. 서 론

최근 무선 통신 시스템에서는 기존의 중앙 데이터 센터를 거쳐서 데이터를 전송 받는 방식에 비해 상대적으로 사용자와 가까운 곳에 컴퓨팅 시스템을 구축해 데이터를 분산처리하는 MEC(Mobile Edge Computing) 기술이 각광받고 있다. 사용자는 중앙 서버와 단말에서 동시에 데이터를 처리할 수 있는 부분 오프로딩을 통해 전체 오프로딩 또는 전체 단말 처리에 비해 가진 자원을 더 효율적으로 사용할 수 있다.

송신 신호에 데이터와 에너지를 통시에 전달할 수 있는 SWIPT (Simultaneous Wireless Information and Power Transmission) 기술 또한 연구가 진행중에 있다. 무선 충전의 특성상 배터리의 교체가 필요 없으며 많은 전력이 필요하지 않은 네트워크에서는 안정적인 에너지원으로 사용될 수 있다. 무선 충전을 위한 실용적인 에너지 추출 회로의 모델에 대한 연구 [1]가 있었으며 SWIPT 를 사용한 MEC 시스템에서 순시적인 관점에서의 지연시간을 최소화하는 연구 또한 진행되었다 [2].

나아가 장기적인 관점에서의 지연시간을 최소화하는 연구 또한 진행되었다 [3]. 장기적인 관점에서는 시간에 따라 바뀌는 채널과 배터리 잔량 등, 변하는 환경 요소들이 고려되어야 하기 때문에 MDP (Markov Decision Process)로 문제를 형성하여 푸는 강화 학습 방법을 사용한다.

위 연구들을 바탕으로 본 논문에서는 SWIPT 가 적용된 MEC 모델에서 부분적 오프로딩을 통한 장기적 관점에서의 지연시간을 최소화하는 문제를 심층 강화 학습을 사용하여 풀 것이다. 본 논문에서는 Value-based 방식에 해당하는 DDQN (Double Deep Q-Network)기법을 사용한다 [4]. DDQN 을 위한 MDP 를 설계한 후 제안한 기법의 성능을 시뮬레이션 결과를 통해 성능을 확인한다.

Ⅱ. 본론

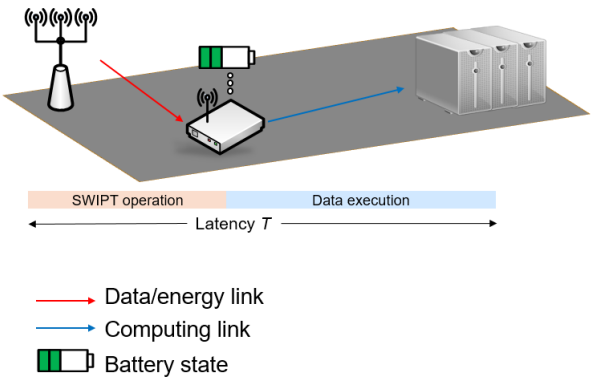

그림 1. 시스템 모델

본 논문에서 사용할 시스템 모델은 단일 사용자 MEC 로 그림 1 과 같다. 사용자는 단말 자체적으로 X_L 만큼의 처리해야할 데이터가 발생하며 AP 로부터 X_C 만큼의 데이터를 SWIPT 시간 T_S 동안 수신함과 동시에 에너지를 추출한다. 단말기에는 비선형적으로 에너지를 추출하는 N 개의 정류기가 포함되어 있어 AP 에서의 송신 전력 P_D 의 일정부분 $1-\rho$ 만큼은 에너지 추출 과정을 거치고 남은 ρ 만큼은 데이터를 수신 받는데 사용된다. 따라서, SWIPT 과정에 걸리는 시간 T_s는 다음과 같다.

$$T_S = \frac{X_C}{\mathrm{B}\log_2\frac{|h_D|^2 P_D}{\sigma^2}\rho} \quad (1)$$

여기서 B, h_D, σ^2 는 각각 채널 주파수 대역, 다운링크 채널 이득, 노이즈 분산 값이다. 또한 T_s 동안 추출되는 전력량은 다음과 같다 [2].

$$P_{har} = \frac{N^* M_{max}}{1-\Omega}\left(\frac{1}{1+\exp(-a(\frac{P_{in}}{N^*}-b))}-\Omega\right) \quad (2)$$

여기서 N^*은 에너지 추출을 최대로 하기 위해 사용되는 정류기의 수, M_{max} 는 정류기당 최대로 추출할 수 있는 전력량, a, b, $\Omega = 1/(1+e^{ab})$은 각 정류기의 특성 마다 사용되는 상수이다.

처리해야하는 총 데이터 양 $X_T = X_L + X_C$ 은 부분적 오프로딩으로 단말 자체적으로 일정부분을 처리함과 동시에 MEC 에 전송해 처리할 수 있다. 일정 비율 β 만큼인 βX_T를 MEC 로 오프로딩하며 나머지 $(1-\beta)X_T$만큼을 자체적으로 처리한다. 그러므로 자체적으로 걸리는 시간 T_{LC} 와 MEC 에 오프로딩 시키는데 걸리는 시간 T_{off}는 다음과 같이 나타낼 수 있다.

$$T_{off} = \frac{\beta X_T}{\mathrm{Blog}_2\left(1+\frac{|h_U|^2}{\sigma^2}P_U\right)}, \qquad T_{LC} = \frac{c(1-\beta)X_T}{f_{LC}} \tag{3}$$

여기서 c, f_{LC}, h_U, P_U는 각각 비트당 요구되는 CPU 회전수, 단말기 자체 CPU 의 computing frequency, 업링크 채널 이득, 단말기의 전송 파워이다. 이에 배터리 모델을 추가하여 추출하는 전력의 일부를 저장할 수 있도록 하였다. 배터리 모델을 사용함으로써 다루고자 하는 지연시간 최소화 문제는 temporal dynamic 하게 되며, 이를 다루고자 model free 한 심층강화학습 중 하나인 DDPG 기법을 사용하였다.

DPPG 를 위한 MDP 는 다음과 같이 설계된다. States 는 $S^k = \left(X^k, H^k, h_D^k, h_U^k\right) \in X = \{0,1\} \times \{(0,\bar{H})\} \times \{h_D\} \times \{h_U\}$ 이며, X^k, H^k는 각각 k 번째 타임슬롯에서 task 발생의 유무와 배터리 잔량이다. Action 은 $A^k = \left(\rho^k, f_{LC}^k, P_U^k\right) \in Y = \{0.1, \cdots, 1\} \times \{0, 0.1, \cdots, 1\} \times \{0, 0.1, \cdots, 1\}$ 이며, f_{LC}^k, P_U^k은 각각 단말의 최대 computing frequency f_{LC}^{max} 와 단말기의 최대 전송 파워 P_U^{max} 의 비율을 나타낸다. Reward 는 풀고자 하는 문제의 목적 함수인 지연시간 $T^k = T_S^k + max(T_{LC}^k, T_{DO}^k)$을 사용한다.

위와 같이 정의된 MDP 를 기반으로 DQN 의 파라미터, $\boldsymbol{\theta}$ 는 $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ 로 업데이트 되며, 여기서 $\alpha, L = (Y_{\text{target}} - Q(S^k, A^k))^2$ 은 각각 학습률과 손실함수를 의미한다. 손실함수는 아래와 같이 정의되는 target action-value 와 action-value $Q(S^k, A^k)$의 평균제곱오차로 정의된다 [4].

$$Y_{\text{target}} = R^{k+1} + \gamma Q(S^{k+1}, \arg\max_a Q(S^{k+1}, a; \boldsymbol{\theta}); \boldsymbol{\theta}_{\text{target}}) \tag{4}$$

시뮬레이션에서 사용한 정류기의 개수는 N=4, 노이즈 분산은 $\sigma^2 = -100$ dBm, 주파수 대역은 B=312.5 kHz 로 설정하였다. 업링크와 다운링크 채널 이득 $|\bar{h}_i|$은 라이시안 factor 3dB 를 가지는 라이시안 분포를 따른다. 비트당 CPU 회전수는 c=100, f_{LC}^{max} 와 P_U^{max} 는 각각 900 MHz, 20 dBm 을 사용하였다. 정류기에 모델에 사용된 상수들은 각각 a=47,083, b=2.9 × 10^{-6}, $M_{max} = 9.079 \times 10^{-6}$ 을 사용하였다. 데이터 양은 $X_c = X_c = 10$ kbits, 오프로딩 비율은 β=0.5, AP 에서 사용자까지의 거리와 사용자와 MEC 서버까지의 거리는 $d_D = d_U = 5$로 설정하였고 AP 의 전송 파워는 P_D=40 dBm 을 이용하였다.

그림 2 는 매 타임슬롯 마다의 loss 가 나타나 있으며 학습이 잘 되어 수렴하는 것을 확인할 수 있다.

그림 3 은 본 논문에서 제시한 모델과 배터리가 없는 모델을 비교한 그림이다. 각 모델마다 task 발생 확률이 1, 0.5, 0.1 인 경우가 그려져 있다. 배터리가 없는 모델에서는 각 타임슬롯에서 추출하는 에너지를 모두 소모하는 경우를 가정하였다. 그림 3 에서 볼 수 있듯이 배터리가 없는 모델은 task 의 발생 빈도와 관계없이 일정한 지연시간에 수렴하는 것을 확인할 수 있다. 제시한 모델에서는 task 의 발생 빈도가 낮아질수록 지연시간이 낮아지는 것을 볼 수 있다.

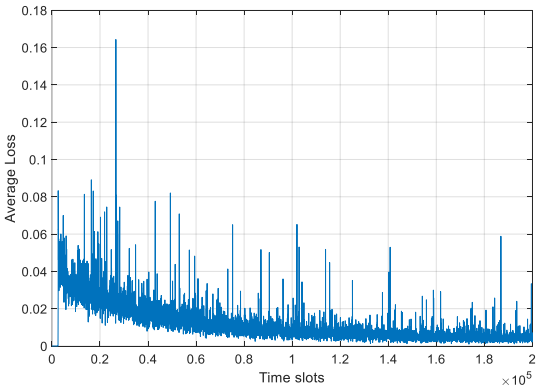

그림 2. Average DQN Loss

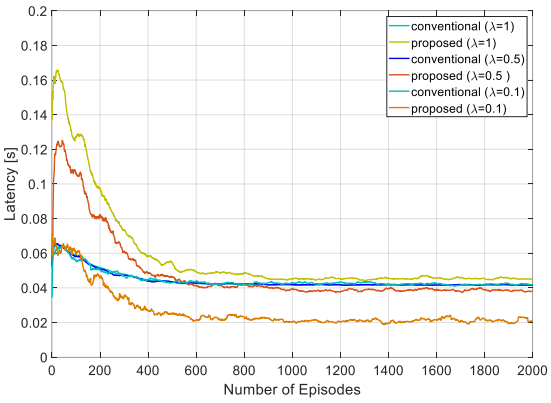

그림 3. 배터리 없는 모델과의 비교

Ⅲ. 결론

본 논문에서는 무선 충전 배터리가 적용된 단일 사용자 MEC 시스템에서 DDQN 강화 학습 방법을 사용하여 장기적 관점에서의 지연시간 최소화 문제를 풀었다. 시뮬레이션을 통해 제안한 기법의 성능을 확인하였다.

ACKNOWLEDGMENT

본 연구는 한국연구재단의 지원을 받아 수행되었음. (No. 2017R1A2B3012316.)

참 고 문 헌

[1] E. Boshkovska, D. W. K. Ng, N. Zlatanov, and R. Schober, “Practical non-linear energy harvesting model and resource allocation for SWIPT systems,” IEEE Commun. Lett., vol. 19, no. 12, pp. 2082- 2085, Dec. 2015.

[2] J. Park, S. Baek, and I. Lee, “Task execution latency optimization for mobile edge computing with energy harvesting,” in Proc. Winter Conference of Korea Information and Communications Society, Feb. 2021.

[3] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, “Performance optimization in mobile-edge computing via deep reinforcement learning,” in Proc. IEEE VTC-Fall, Chicago, IL, USA, Aug. 2018, pp. 1- 6.

[4] H. van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-Learning”, AAAI, vol. 30, no. 1, Mar. 2016.