

IIoT 시나리오에서 신뢰성을 보장하는 에너지 효율적인 오프로딩

구설원, 임유진
숙명여자대학교 IT 공학과
tjrgkr501@sookmyung.ac.kr, yujin91@sookmyung.ac.kr

Energy-efficient offloading to ensure reliability in IIoT scenarios

Seolwon Koo, Yujin Lim
Dept. of IT Engineering, Sookmyung Women's University

요 약

Mobile Edge Computing(MEC)는 사용자 근처에서 서비스를 제공하기 때문에 사물인터넷에서 주목받고 있는 기술이다. 오프로딩을 통한 MEC 서버의 활용은 제한된 배터리 수명이나 계산 능력을 갖는 디바이스들에게 매우 유용하다. 본 논문은 강한 신뢰도가 요구되는 산업 사물인터넷(Industrial IoT, IIoT) 시나리오를 가정하여, 태스크를 실행할 때 발생하는 에너지 소모량과 지연 시간을 최적화하며 신뢰도를 보장하는 오프로딩 기법을 제시한다. 본 연구는 실험을 통해 에너지 소모량과 신뢰성 측면에서 제안 기법의 성능을 분석하였다.

1. 서론

사물인터넷(Internet of Things, IoT)의 발전으로 수많은 모바일 디바이스가 활용되고 있고, 다양한 서비스 제공을 위해 모바일 디바이스들은 많은 계산을 요구하는 태스크들을 발생시키고 있다. 모바일 디바이스의 배터리 수명과 계산 능력은 제한되어 있기 때문에 이를 해결하기 위해서 클라우드 컴퓨팅 환경을 사용했다. 하지만 모바일 디바이스와 클라우드 데이터 센터는 지리적으로 멀어 지연시간에 제한 조건이 있는 응용 서비스에 적합하지 않다. 이 문제를 해결하기 위해 MEC 개념이 도입되었다. MEC는 분산 방식으로 모바일 디바이스 가까이에서 위치하며 저장과 계산 기능을 제공한다. 모바일 디바이스에서 MEC 서버를 사용하기 위해서는 계산 오프로딩(computation offloading)을 수행해야 한다.

계산 오프로딩은 디바이스의 태스크를 MEC 서버로 전송하여 처리하는 것을 의미한다. 이는 전체 오프로딩(full offloading)과 부분 오프로딩(partial offloading)으로 나뉘어진다. 전체 오프로딩은 태스크 전체를 MEC 서버로 전송하는 것을 의미하며, 부분 오프로딩은 태스크의 일부분을 MEC 서버로 전송하여 처리하는 것을 의미한다. 현재 다양한 조건과 요구

사항을 고려하여 최적화된 오프로딩 결정에 대한 연구가 많이 진행되고 있다.

기존 오프로딩에 대한 연구[1]은 매칭 이론을 이용하여 처리 지연시간을 최소화하는 오프로딩 기법을 제시하였고, [2]는 DAG(Directed Acyclic Graph)를 기반으로 태스크 종속성을 보장하면서 신뢰성과 지연시간의 제약을 만족시키는 에너지 최적화 오프로딩 결정 기법을 제시하였다.

IoT는 유비쿼터스 컴퓨팅 특징으로 인해 많은 도메인에서 이용되고 있다. 그중에서도 산업 도메인에 활용되는 IIoT는 기존 IoT와 다르게 신뢰성(reliability)에 있어서 보다 강력한 요구 사항을 가지며, 태스크의 양이 상대적으로 크다. IIoT 환경에서는 앞서 언급한 대로 신뢰성이 증시되는데 그 이유는 다른 도메인과 비교하였을 때 많은 돈이 투자되고 신뢰성이 떨어지게 되면 손실액이 다른 도메인에 비해 크기 때문이다. 또한 IoT 모바일 디바이스의 배터리 수명과 계산 능력이 제한적이라는 점도 여전히 존재한다[3].

본 연구에서는 IIoT 환경에서 모바일 디바이스의 오프로딩 결정 기법을 제안하고자 한다. IIoT 환경에서 신뢰성을 보장하기 위해서 모바일 디바이스의 에

너지 소모와 지연시간을 동시에 최소화할 수 있도록 하여 모바일 디바이스의 제한적인 배터리 수명과 계산 자원 문제를 해결하고자 한다. 다시 말해서 지연 제약 조건을 만족시키면서도, 모바일 디바이스의 에너지 소모를 최소화하는 오프로딩 결정 기법을 제안한다.

2. 시스템 모델

본 연구는 여러 디바이스와 한 개의 MEC 서버 환경을 가정하며, 디바이스들과 MEC 서버는 무선 네트워크로 연결되어 있다고 가정하였다. 그리고 디바이스 하나당 하나의 태스크가 발생하고, 전체 오프로딩만을 가정하였다. 태스크의 집합은 $T = \{t_1, t_2, \dots, t_n\}$ 로 표기하며, 각 태스크 별로 $t_n = \{w_n, u_n\}$ 의 2개의 속성을 갖는다. w_n 는 발생된 태스크의 크기(KB)이며, u_n 는 태스크 실행을 완료하기 위해 요구되는 프로세서 사이클(Megacycles)의 개수이다. L_{max} 는 태스크의 최대 지연 제약이다. 통신 모델은 Shannon-Hartley 식을 이용하였다. R_n 는 디바이스 n 과 MEC 서버 사이의 데이터 전송률이다.

$$R_n = \frac{B}{N} \cdot \log_2 \left(1 + \frac{p_n^{tra} \cdot h^2}{\sigma^2} \right) \quad (1)$$

B 는 전송 채널 대역폭이고, N 는 디바이스 개수이다. σ^2 는 잡음 이득이며, h^2 는 평균 경로 손실을 의미한다. p_n^{tra} 는 t_n 의 전송 전력이다. 본 연구에서는 실행이 완료된 태스크의 출력 데이터가 입력 데이터에 비해 상대적으로 매우 작기 때문에 출력 데이터의 전송 지연시간은 무시하였다.

t_n 를 로컬에서 처리할 때의 지연시간(L_n^l)은 다음과 같다.

$$L_n^l = \frac{u_n}{f_n} \quad (2)$$

f_n 는 디바이스의 계산 능력이다. t_n 를 로컬에서 처리할 때의 디바이스 에너지 소모량(E_n^l)은 다음과 같다.

$$E_n^l = \kappa \cdot (f_n)^2 \cdot u_n \quad (3)$$

κ 는 디바이스에 의해서 결정되는 상수이다. 태스크 t_n 를 MEC 서버로 오프로딩하여 실행할 때 발생하는 지연시간은 디바이스에서 MEC 서버까지의 태스크 전송 지연시간에 MEC 서버에서의 태스크 처리 지연시간을 더한 값과 같다.

$$L_n^{off} = \frac{w_n}{R_n} + \frac{u_n}{f_{MEC}} \quad (4)$$

f_{MEC} 는 MEC 서버의 계산 능력이고, t_n 를 MEC 서버로 보냈을 때 디바이스가 소모하는 에너지는 다음과 같다.

$$E_n^{off} = p_n^{tra} \cdot \left\{ \frac{w_n}{R_n} \right\} + p_n^{idle} \cdot \left\{ \frac{u_n}{f_{MEC}} \right\} \quad (5)$$

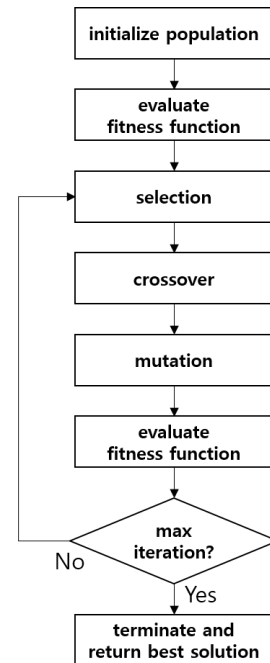
MEC 서버로 t_n 를 전송할 때 소모되는 에너지와 MEC 서버에서 t_n 를 처리할 동안 소모되는 에너지를 더한

값이다. p_n^{idle} 는 디바이스 n 의 유휴전력을 의미한다. MEC 서버는 유선 전력망에 의해서 컴퓨팅 에너지가 제공되기 때문에 MEC 서버의 에너지 소모량은 고려하지 않았다.

본 연구의 목적은 지연 제약조건(L_{max})을 만족하면서 태스크 t_n 의 처리 지연시간(L_n)과 디바이스 에너지 소모량(E_n)을 동시에 최소화하는 것이다. 이를 위하여 본 연구에서는 유전 알고리즘을 사용하였다.

3. 제안 알고리즘

본 연구는 MEC 서버가 디바이스와 태스크 정보를 기반으로 유전 알고리즘을 이용하여 태스크의 오프로딩 여부를 결정하도록 하였다. 본 연구에서는 유전자(gene)가 0과 1로만 이루어진 이진 유전 알고리즘을 사용한다. 전반적인 유전 알고리즘의 과정은 그림 1을 통해 볼 수 있다.



(그림 1) 유전 알고리즘 순서도

하나의 염색체(chromosome)는 모든 태스크들의 오프로딩 여부를 나타내는 배열을 의미하며, 선택(selection)은 룰렛 휠 선택을 사용했고, 교배(crossover)는 single point 를 이용했다. 그리고 돌연변이(mutation)은 reverse 를 이용했다.

$$\text{fitness} = 1 / (\alpha \cdot E^{nor} + (1 - \alpha) \cdot L^{nor}) \quad (6)$$

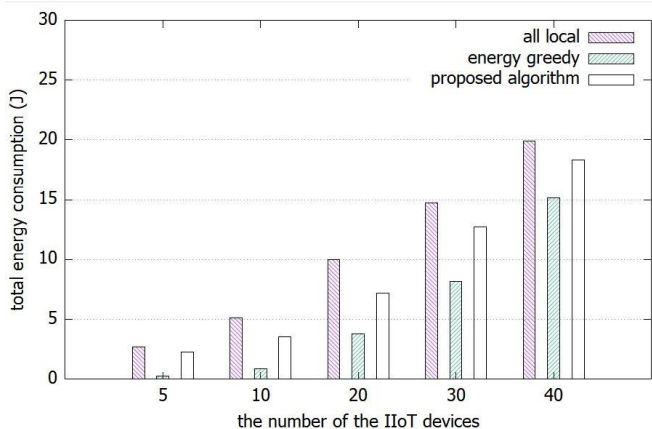
적합도 함수는 식(6)으로 정의했다. 적합도 함수는 가중 합을 이용하여 정의했고, α 는 0과 1사이의 가중치 값이다. E^{nor} 는 모든 태스크의 전체 에너지 소모량을 정규화한 값이고 L^{nor} 는 모든 태스크의 전체 지연시간을 정규화한 값이다.

4. 실험 및 결과 분석

제안한 알고리즘의 성능 평가를 위해 다음과 같은 실험을 진행하였다. B는 10MHz, h^2 는 10^{-6} 그리고 σ^2 는 10^{-9} W이다. κ 는 10^{-27} , p_n^{tra} 과 p_n^{idle} 는 각각 0.1W, 0.001W이다. 디바이스에서 발생하는 태스크의 사이즈 (w_n)와 CPU 사이클(u_n)는 각기 [600,1000]KB 와 [400,600]Megacycles로 설정하였다. 그리고 디바이스와 MEC 서버의 계산 능력인 f_n 과 f_{MEC} 는 각각 1GHz와 5GHz이다. 지연 제약조건(L_{max})는 0.5sec이다. 그리고 유전 알고리즘에서 사용된 세부 설정은 다음과 같다. 교배 확률은 0.8, 돌연변이 확률은 0.2로 설정했다. 그리고 모집단(population) 크기는 100으로 설정했으며, 반복 횟수는 1000이다. α 는 0.5이다.

제안한 알고리즘의 성능 평가를 위해 2가지의 기법과 비교하면서 실험을 진행하였다. 첫 번째 기법은 모든 태스크를 로컬로 처리하는 all local 기법이고, 두 번째 기법은 지연 제약조건을 무시하고 오프로딩 결정 값 중 에너지가 소모량이 제일 작은 것으로 선택하는 energy 기반 greedy 기법이다.

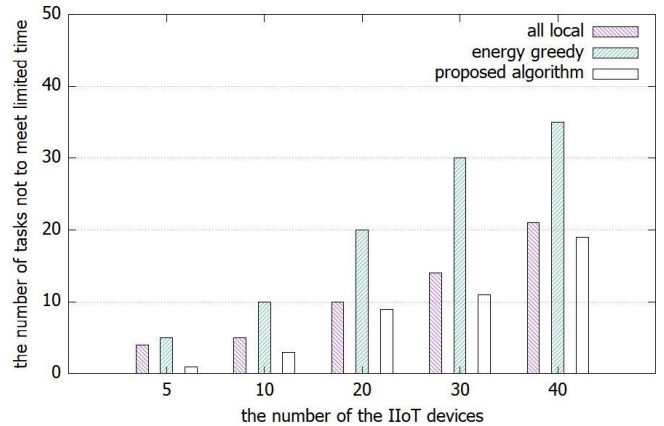
그림 2는 디바이스 수에 따른 전체 태스크의 에너지 소모량에 대한 그래프이다. Energy greedy 기법의 경우 전체 에너지 소모량 최소화가 목적이기 때문에 결과가 가장 좋다. 본 연구에서 제안하는 알고리즘은 all local 기법과 비교하여 전체 에너지 소모량에 있어서 19% 정도로 좋은 성능을 보였다. 그리고 energy greedy 기법과의 성능 비교에서 energy greedy가 제안 알고리즘보다 약 52% 정도 좋았다.



(그림 2) 디바이스 수에 따른 전체 에너지 소모량

그림 3은 디바이스 수에 따른 지연 제약 조건을 만족시키지 못한 태스크 수이다. 그림 3의 성능 비교는 신뢰성의 척도가 된다. 제안 알고리즘과 비교하였을 때 all local에 비해 약 31% 정도 좋은 성능을 보였다. Energy greedy의 경우 오직 에너지 소모량만을 최적화했기 때문에 제안 알고리즘이 약 62% 정도 더 나은 성능을 보였다. Energy greedy 기법의 경우 전체 에너지 소모량에서 월등하게 좋은 성능 결과를 보

였지만 월등하게 좋았던 만큼 신뢰도에 있어서는 안 좋은 성능을 보였다. 신뢰도에서 안 좋은 성능을 보였다는 의미는 지연 제약조건을 만족하지 못해 실패한 태스크가 많았다는 것을 나타낸다. 총 에너지 소모량의 효율이 좋아도 신뢰도가 좋지 않으면, 신뢰성을 중시하는 IIoT 환경에서 적합하지 않기 때문에 energy greedy 방법이 성능이 효율적이라고 볼 수 없다. 결과적으로 본 연구는 다른 연구들에 비해 성공한 태스크 수가 많아 신뢰도를 보장하면서 전체 에너지 소모량을 최소화했다는 것을 알 수 있다.



(그림 3) 디바이스 수에 따른 지연 제약조건을 만족시키지 못한 태스크 수

5. 결론 및 향후 연구

본 연구에서는 IIoT 환경에서 유전 알고리즘을 이용하여 오프로딩 여부를 결정하는 기법을 제안하였다. 이 알고리즘이 다른 두 기법과 비교하여 에너지 소모량과 신뢰성에 있어서 좋은 성능을 보였다.

향후 연구에서는 다수의 MEC 서버 간 로드밸런싱을 고려하여 에너지와 지연시간을 최적화하는 연구를 진행하고자 한다.

사사문구

이 성과는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2018R1A2B6002505)

참고문헌

[1] F. Chiti, R. Fantacci and B. Picano, "A Matching Theory Framework for Tasks Offloading in Fog Computing for IoT Systems," IEEE Internet of Things Journal, vol. 5, no. 6, pp. 5089-5096, 2018

[2] H. Liu, L. Cao, T. Pei, Q. Deng and J. Zhu, "A Fast Algorithm for Energy-Saving Offloading with Reliability and Latency Requirements in

- Multi-Access Edge Computing," IEEE Access, vol. 8, pp. 151-161, 2020
- [3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4724-4734, 2018