

# 커넥티드 머신을 위한 통신-컴퓨팅 융합 기술

이 병 주\*, 노 정 훈°

## Communication-Computing Convergence Technique for Connected Machines

Byungju Lee\*, Jung-Hoon Noh°

### 요 약

디지털 트윈, 메타버스와 같은 몰입형 서비스를 지원하기 위해서는 높은 컴퓨팅 성능과 에너지 소모가 필요하다. 일반적으로, 단말은 데스크톱 PC 및 서버에 비해 상대적으로 낮은 컴퓨팅 성능을 가지고 있어 이러한 몰입형 서비스를 지원하기 위해서는 컴퓨팅 오프로딩은 필수적이다. 본 논문에서는 실제 통신 표준 오버헤드를 고려하는 컴퓨팅 오프로딩 기법을 제안한다. 우선, 제한한 컴퓨팅 오프로딩 기법에서 고려하는 오프로딩 절차 및 요구 조건들을 설명한다. 실험 결과를 통해 실제적인 조건을 반영하는 의사 결정 알고리즘이 지연 시간 감소 및 에너지 소비 효율성을 향상시킬 수 있음을 보인다.

**Key Words** : Immersive Services, Computation Offloading, Artificial Intelligence, Deep Q-learning, Computation Return

### ABSTRACT

Evolving immersive services such as digital twin and metaverse require a vast amount of computing power and extreme energy consumption. In order to support such immersive services with a mobile device which in general has much less computing

power than a desktop PC and server, offloading of the computation is essential. In this paper, we propose a computation offloading algorithm considering practical overhead and power consumption of the wireless network standards. From numerical evaluation, we show that the proposed decision making algorithm can decrease the latency and improve the energy consumption efficiency.

### I. 서 론

5세대 이동통신서비스(5G)가 2019년 상용화된 후, 6세대 이동통신서비스(6G) 연구가 활발히 진행되고 있다. 6G는 2030년경에 상용화 될 것으로 예상되고 있고 그 무렵에는 5000억개의 머신들이 네트워크에 연결될 것으로 전망된다<sup>[1]</sup>. 이러한 커넥티드 머신의 예로는 홀로그램 기기, 드론, 차량, 로봇 등이 있을 수 있다.

디지털 트윈, 메타버스와 같은 몰입형 서비스의 등장으로 컴퓨팅 오버헤드는 중단간 지연 시간, 에너지 소비 및 메모리 오버헤드를 크게 증가시킬 수 있다. 특히, AR(Augmented Reality) 글래스, VR(Virtual Reality) 헤드셋과 같은 경량 기기들에는 심각한 전력 소모를 유발하고 결국 모바일 장치의 배터리 수명을 단축시킬 것이다. 모바일 기기의 부족한 컴퓨팅 성능을 보완하기 위해 모바일 에지 컴퓨팅(MEC, Mobile Edge Computing)이 학계와 산업계 모두에서 많은 관심을 받고 있다<sup>[2]</sup>. MEC는 모바일 기기의 계산 작업을 근처의 에지 서버로 오프로딩하고 이를 통해 연산 지연과 배터리 소모를 줄일 수 있다.

기존 연구들을 살펴보면, 게임 이론 및 반복 알고리즘 기반으로 오프로딩 결정과 무선 대역폭 할당의 공동 최적화를 위한 방법들이 제시되었다<sup>[3][4]</sup>. 또한, 오프로딩 결정을 위해 다중 병렬 신경망을 활용하는 분산 딥러닝 기반 알고리즘이 제안된 바 있다<sup>[1]</sup>. 기존 연구들에서 주목할 점은 실제 통신시스템의 헤더 및 스케줄링 신호와 같은 페이로드 전달에 필요한 실제 오버헤드가 고려된 바 없다는 것이다. 이 경우, 서로 다른 업무의 지연 요건으로 인해 MEC의 문제가 더욱

\* 이 연구는 금오공과대학교 학술연구비로 지원되었음(202003700001).

• First Author : (ORCID:0000-0002-1710-2324)Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering, byungju@kumoh.ac.kr, 조교수, 종신회원

° Corresponding Author : (ORCID:0000-0002-4599-0584)Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering, jhnoh@kumoh.ac.kr, 조교수, 정회원

논문번호 : 202201-005-A-LU, Received January 5, 2022, Revised February 9, 2022; Accepted February 19, 2022

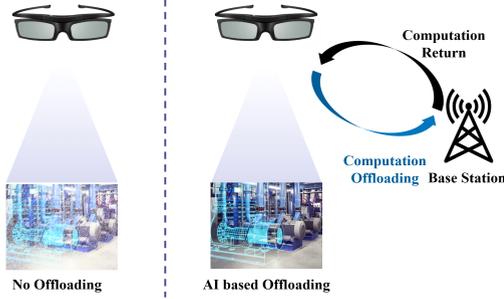


그림 1. 몰입형 서비스 지원을 위한 AI 기반 오프로딩  
 Fig. 1. AI based offloading to support immersive services

어려워지게 된다.

본 논문에서는, 무선 통신 시스템의 실질적인 오버헤드를 고려하는 AI(Artificial Intelligence) 기반 오프로딩 기법을 제안한다. 제안된 방법의 주요 특징은 지연 요구 사항(예를 들어, AR 이미지 처리), 실제 오버헤드, 서버 정보 등을 모두 고려하여 애플리케이션을 처리하는 것이다.

## II. 시스템 모델 및 오프로딩 결정 알고리즘

본 논문에서 고려하는 통신-컴퓨팅 융합 시나리오에서는 처리 시간 정보를 바탕으로 단말에서의 처리 시간이 서버에서의 처리 시간보다 긴 경우 단말이 오프로딩을 하는 시스템을 고려한다. 커넥티드 단말들의 오프로딩 결정 최적화 문제는 다음과 같이 모델링할 수 있다.

$$\begin{aligned}
 & \underset{x}{\operatorname{argmin}} \sum_{i=1}^N [x_i t_i^M + (1-x_i) t_i^E] \\
 & \text{s.t.} \sum_{i=1}^N [x_i P_i^M + (1-x_i)(P_i^{Tx} t_{Tx} + P_i^{Rx} t_{Rx})] < P_{\max}^M \\
 & \sum_{i=1}^N [(1-x_i) P_i^{MEC} t_i^{MEC}] < P_{\max}^{MEC} \\
 & \sum_{i=1}^N [x_i t_i^M + (1-x_i) t_i^E] < T_{QoS} \\
 & \sum_{i=1}^N (1-x_i) f_i^{MEC} < f_{\max}^{MEC}
 \end{aligned} \tag{1}$$

여기서  $X = [x_1, x_2, \dots, x_N]$ 는 전체 단말들의 오프로딩 결정이고  $x_i \in \{0, 1\}$ 는  $i$ 번째 단말의 오프로딩 결정이다.  $x_i = 0$ 일 때는, 계산이 단말에서 수행되고,  $x_i = 1$ 일 때는, 계산이 에지 서버에서 수행된다. 그리고 단말에서의 처리 시간은  $t_i^M = d_i / f_M$ 와 같다. 여기서  $d_i$ 는  $i$ 번째 단말의 계산을 위한 데이터이고  $f_M$ 은

단말의 컴퓨팅 속도이다. 계산 작업이 에지 서버에서 처리되는 경우의 처리시간은  $t_i^E = t_{Tx} + t_i^{MEC} + t_{Rx}$ 와 같다. 여기서  $t_{Tx}, t_{Rx}$ 는 계산할 데이터를 전송하는 시간과 계산된 결과를 수신하는 시간이다.  $t_i^{MEC} = d_i / f_i^{MEC}$ 는 에지 서버의 처리 시간이고  $f_i^{MEC}$ 는  $i$ 번째 단말을 위한 에지 네트워크의 컴퓨팅 속도이다. 데이터 크기가 작을 때는, 실제 송수신 시간이  $d_i / Ch_{rate}$ 와 같이 단순하게 표현되지 않는다. 여기서  $Ch_{rate}$ 는 송신 또는 수신 전송률 ( $Ch_{Tx}$  또는  $Ch_{Rx}$ )이다. 실제 시스템에서 데이터는 헤더, 테일과 함께 패키징이 되고, 또한 스케줄링 요청 메시지, 스케줄링 확인 메시지와 같은 사전 시그널링이 필요하다. 이와 같은 오버헤드를 고려하는 경우의 데이터 전송 시간은  $t_{Tx} = (d_i + Hdr) / Ch_{Tx} + t_{Tx}^{ov}$ 와 같이 되고 여기서  $Hdr$ 은 헤더 사이즈, 그리고  $t_{Tx}^{ov}$ 는 추가 시그널링을 위해 소요되는 시간이다. 에지 서버에서 계산된 결과를 수신하는 시간은  $t_{Rx} = (d_i^{Rx} + Hdr) / Ch_{Rx} + t_{Rx}^{ov}$ 와 같고  $d_i^{Rx}$ 는 계산된 데이터 사이즈이고  $t_{Rx}^{ov}$ 는 추가 시그널링을 위해 소요되는 시간이다.

단말의 전력 소비량은  $P_i^M = P_i^C t_i^M$ 와 같이 나타낼 수 있고, 여기서  $P_i^C$ 는  $i$ 번째 사용자의 컴퓨팅을 위한 파워 전력이다. 오프로딩 전력 소비량은  $P_i^{Tx} t_{Tx} + P_i^{Rx} t_{Rx}$ 와 같이 나타나고 여기서  $P_i^{Tx}$ 와  $P_i^{Rx}$ 는 각각 송신 파워와 수신 파워이다.  $P_i^{MEC}$ 는  $i$ 번째 사용자를 위한 에지 서버의 전력 소비량이다.  $T_{QoS}$ 는 단말의 QoS를 만족하기 위한 최대 허용 지연 시간이다.

그리고 앞서 설명한 시스템 모델에서 서버의 컴퓨팅 역량 정보, 서버의 용량 정보, 서버의 통신 채널 상황에 대한 정보를 바탕으로 단말의 오프로딩할 데이터를 분할 할 수 있는 경우에는 그림 2와 같은 오프로딩 절차로 확장된다.

또한, 위의 다양한 정보를 고려하는 경우, Deep Q-learning 기반 오프로딩 결정 알고리즘을 적용할 수 있다. 본 논문에서 설명한 시스템 모델을 적용하면

$$\begin{aligned}
 & \text{state는} \quad S = \sum_{i=1}^N [x_i t_i^M + (1-x_i) t_i^E], \quad \text{action은} \\
 & X = [x_1, x_2, \dots, x_N], \quad \text{reward는} \quad (\max_{i=1}^N [x_i t_i^M + \\
 & (1-x_i) t_i^E] - Q(s, x)) / (\max_{i=1}^N [x_i t_i^M + (1-x_i) t_i^E]) \text{가 된다.}
 \end{aligned}$$

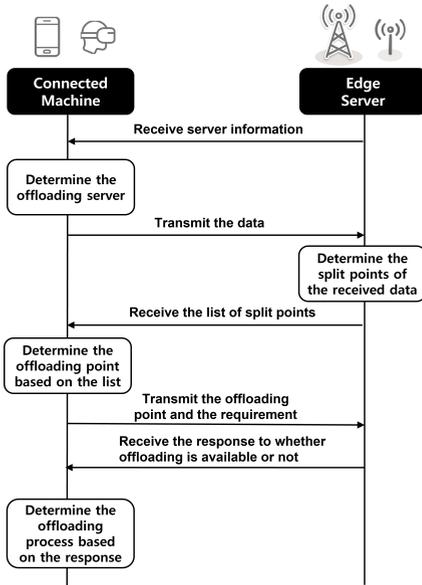


그림 2. 단말의 오프로딩 의사 결정 절차  
Fig. 2. Offloading decision making procedure at the connected machine

### III. 실험결과 및 토의

본 논문에서는 실험 결과를 통해 제안한 AI 기반 알고리즘이 지연 시간 감소와 에너지 효율 향상 관점에서 효과적인 기법임을 보인다. 그림 3은 단말들이 모든 연산을 처리하는 경우(No Offloading), 에지 서버에서 모든 연산을 처리하는 경우(Full Offloading), 무선 송수신(Transmission+Reception), 제안 기법(AI based Offloading)들에 대해 작업 크기에 따른 처리 시간을 나타내었다. 송수신 채널 50Mbps, 에지 서버

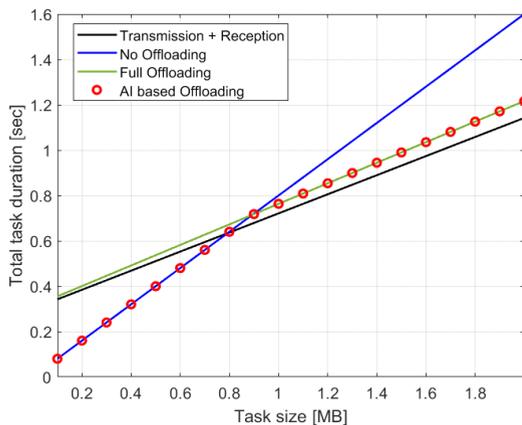


그림 3. 데이터 크기에 따른 작업 소요 시간  
Fig. 3. Total task duration with respect to task size

대 단말의 컴퓨팅 역량 비율 20:1을 고려하였다. 실제적인 오버헤드도 고려하여, 페이로드가 80 비트인 경우, 헤더 및 기타 오버헤드를 고려하면 실제 패킷 길이는 약 750 비트가 된다. 실험 결과를 통해 데이터 크기가 작은 경우에는 실제적인 무선 송수신 시그널링과 패킷 패키징 오버헤드 때문에 오프로딩이 효율적이지 않음을 알 수 있다. AI 기반 알고리즘은 변조 및 코딩 기술, 채널 품질, 대역폭, 에지 서버까지의 라우팅 시간, 데이터 크기, 단말들의 컴퓨팅 능력, 에지 서버의 컴퓨팅 능력에 따라 어떤 작업을 오프로딩 할지 결정하게 된다.

그림 4에는 단말 수에 따른 에너지 소비량을 나타내었다. 단말의 에너지 소비량은  $e_i^M = \rho^M \omega_i$ 와 같고, 여기서  $\rho^M$ 는 CPU 사이클 당 에너지 소비 계수이고,  $\omega_i$ 는  $i$ 번째 작업을 완료하는 데 필요한 CPU 사이클 수이다. 에지 서버로 오프로딩하여 처리하는 에너지 소비량은  $e_i^E = \rho^E \omega_i + (\rho_i^{Tx} d_i) / Ch_{Tx}$ 이다. 여기서  $\rho_i^{Tx}$ 는 데이터를 전송하는데 드는 파워이다.  $\omega_i$ 는 1GHz와 5GHz 사이의 값,  $d_i$ 는 10MB와 50MB 사이의 값을 갖고,  $\rho^E (< \rho^M) = 10^{-9}$ ,  $\rho_i^{Tx}$ 의 최대값은 0.1W이다. 그림 4를 통해 오프로딩을 하지 않는 경우의 에너지 소비량이 다른 두 경우의 에너지 소비량보다 더 높은 것을 확인할 수 있다. 제안한 기법은 작은 사이즈 작업을 에지 서버로 오프로딩하지 않으므로써 전체 오프로딩에 비해 전체 에너지 소비량을 더욱 줄일 수 있다는 점에 주목할 필요가 있다. 그 이유는, 작은 사이즈 작업에 소비되는 에너지 중 단말의 컴퓨팅

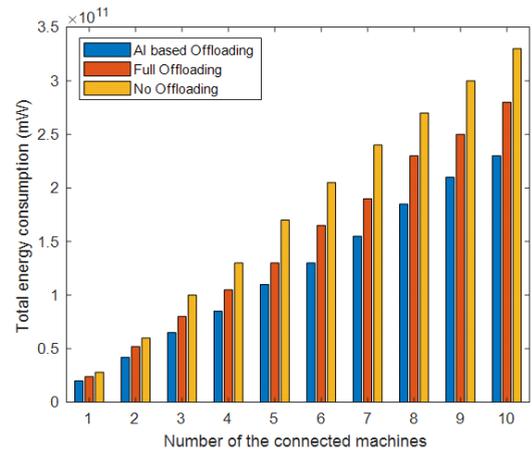


그림 4. 단말 수에 따른 전체 에너지 소비량  
Fig. 4. Total energy consumption comparison with respect to the number of connected machines

이외의 데이터 전송이 지배적인 부분을 차지하기 때문이다.

다가오는 6G 시대에서는 네트워크 밀도가 증가함에 따라, 기지국, 액세스 포인트, 단말의 컴퓨팅 자원까지 활용할 수 있을 것으로 예상된다. 이를 위해서는, 코드화된 분산 컴퓨팅, 연합 학습, 분할 학습 등이 고려되어야 할 것이다.

## References

- [1] Samsung Research, *6G White Paper: The Next Hyper-Connected Experience for All*, Jul. 2020, from <https://research.samsung.com/>
- [2] E. Lee and S. Lee, "Task offloading algorithm for mobile edge computing," *J. KICS*, vol. 46, no. 2, pp. 310-313, Feb. 2021.
- [3] J. Xue and Y. An, "Joint task offloading and resource allocation for multi-task multi-server NOMA-MEC networks," *IEEE Access*, vol. 9, pp. 16152-16163, 2021.
- [4] J. Park, S. Baek, and I. Lee, "Task execution latency optimization for mobile edge computing with energy harvesting," in *Proc. KICS Winter Conf. 2021*, pp. 652-653, Feb. 2021.