# 모바일 에지 컴퓨팅에서 에너지 최적화를 위한 다중 장치 작업 오프로드 프레임워크


당꽝녓밍, 유명식
minhdqn@soongsil.ac.kr, myoo@ssu.ac.kr


# A Multi-device Task Offloading Framework For Energy Optimization In Mobile Edge Computing


Dang Quang Nhat Minh, Myungsik Yoo



## Abstract

The fast development of mobile-based technologies has utilized resource-hungry mobile applications, such as Augmented Reality (AR), Virtual Reality (VR), vehicular networks, etc. However, the fast development of these applications leads to more complexity and requires more computation resources over time. Mobile Edge Computing (MEC) is introduced that brings computation environment near the end-users to achieve essential requirements such as low energy consumption, low latency, low cost. This paper proposes a task offloading framework to optimize the energy consumption of multi-device in consideration of the edge server computation capacity in MEC environment.


## Ⅰ. Introduction

With the rapid development of mobile applications, there has been a great striving for computation capacity to provide various services. Furthermore, the massive amount of data generated by those applications cause some significant problems such as security, delay, energy consumption, etc [1]. Mobile devices have limited computation capacity and energy, so some massive tasks can not run locally [2,3]. Mobile Edge Computing (MEC) is introduced to handle these challenges. It brings the computation environment near the end-users [4].

Mobile can offload the task to edge cloud for computing or choose to execute it in the local [5]. In most existing work on task offloading, the edge cloud computation power value has been already known and been a fixed value. However, the computation capacity can change overtimes dynamically because of various kinds of computation tasks. And this will affect the offloading decision of mobile devices.

In this paper, we formulate and propose a multi-device task offloading framework for energy optimization based on the computation capacity of the edge cloud.

## Ⅱ. Proposed system model

Figure 1 shows the system model. There are multiple smart devices, edge cloud servers, and a central cloud server in the system. The mobiles send the energy and task information to the closest edge servers. There are multiple edge servers $R = \{1,2,\dots,R\}$ in the system. The edge servers have bigger computation capacities for computing heavy tasks. Suppose there are N smart devices and M multiple tasks. Then, we describe the set of multiple devices as $N = \{1,2,\dots,N\}$; and set of multiple tasks as $M = \{1,2,\dots,M\}$.

We suppose that a mobile device can only request one task per time slot, and mobile devices are fixed in the system. We define:

- The task m that the mobile n request as $u_{n,m}$
- The computation resource required for $u_{n,m}$ is denoted as $r_{n,m}$
- The total data size for offloading is $s_{n,m}$
- The required completion time is $d_{n,m}$.
- The computation capacity of the edge server is $c_{n,m}$

### A. Communication model

The uplink data transfer communication is described in Equation (1) below:

$$trans\_up_n = Bandwidth * \log_2\left(1 + \frac{P_n G_n}{\sigma^2 + I_n}\right) \quad (1)$$

Where $Bandwidth$ is the channel bandwidth. We denote $P_n$ as the mobile $n$'s transmitted power, $G_n$ as the gain of the channel between the edge cloud and mobile $n$. $I_n$ is the intervention between edge cloud and mobile $n$, and $\sigma^2$ indicates the Gaussian noise power.

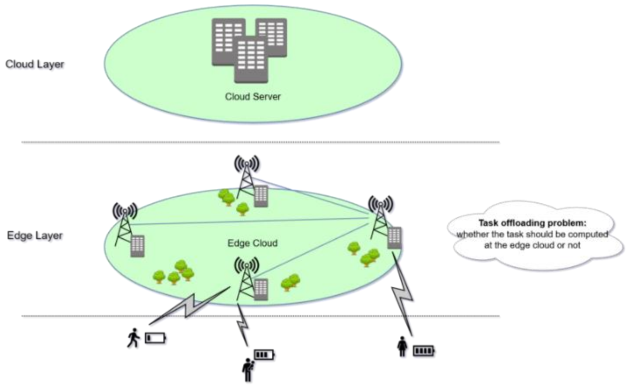


Figure 1 Proposed multi-device task offloading system

The same with studies [5-8], we ignore the lag of downlink transmission due to many applications' output data is typically substantially smaller than their incoming data.

B. Computation model

- Mobile device model

$$Energy^{l}_{n,m} = r_m \beta^l \qquad (2)$$

The computation energy utilization of mobile device $n$ is given in Equation 2. We denote $\beta^l$ as the coefficient. According to study [5], the $\beta^l$ is expressed in Equation below:

$$\beta^l = k(f^l_n) \qquad (3)$$

Where $k$ is the energy coefficient and equals to $10^{11}$ according to [9]. We denote $f^l_n$ as the computation resource of the mobile device $n$.

The total computation latency when executing the task $m$ in mobile device $n$ is express as:

$$T^{l}_{n,m} = \frac{r_n}{f^l_n} \qquad (4)$$

- Mobile edge server model

The communication delay has three parts: (1) the task process time $T^{proc_{edge}}_{n,m}$, (2) the transmission time $T^{tra_{edge}}_{n,m}$, and (3) the queue delay $T^{queue_{edge}}_{n,m}$. The communication delay is obtained by:

$$T^{edge}_{n,m} = T^{proc_{edge}}_{n,m} + T^{tra_{edge}}_{n,m} + T^{queue_{edge}}_{n,m} = \frac{s_{n,m}}{Energy^{l}_{n,m}} + \frac{r_{n,m}}{f^{edge}_{n,m}} + T^{queue_{edge}}_{n,m} \qquad (5)$$

where $f^{edge}_{n,m}$ is the CPU computation resource of the edge server assigned to $u_{n,m}$. The energy utilization of mobile device $n$ that sends the task $m$ to the edge server can be obtained by:

$$Energy^{edge}_{n,m} = P_n T^{tra_{edge}}_{n,m} \qquad (6)$$

- Problem statement

In this study, the task offloading decision is a binary option. We define $h_{n,m} \in \{0,1\}$, where the value 0 indicates that the task is executed at mobile device and value 1 indicates that the task is executed at the edge cloud.

The objective of this study is to minimize the energy consumption of mobile devices in consideration of the CPU computation capacity of the edge cloud, which can be expressed below:

$$\min_{h_{n,m}} \sum_{n=1}^{N} Energy_{n,m}\left(Energy^{l}_{n,m}, Energy^{edge}_{n,m}\right) \qquad (7)$$

We also present four constraints as below:

$$\begin{aligned} &Constraint\ 1\text{: } T_{n,m} < d_{n,m}, \quad \forall n,m \in N,M \\ &Constraint\ 2\text{: } \sum_{r=1}^{R} u_{n,m} \le 1, \quad \forall n,m \in N,M \\ &Constraint\ 3\text{: } \sum_{n=1}^{N} h_{n,m}\, f^{edge}_{n,m}, \quad \forall m \in M \\ &Constraint\ 4\text{: } h_{n,m} \in \{0,1\} \end{aligned} \qquad (8)$$

Constraint 1 guarantees that the task $m$ must be completed before the required time. Constraint 2 states that each mobile device $n$ can connect and offload the task $m$ at most one edge server. Constraint 3 presents the computation utilization needed for executing the tasks must be smaller than the computation capacity of edge cloud at each time slot. And finally, Constraint 4 mentions the offloading decision is a binary option.

## Ⅲ. Conclusions

In MEC system, mobile devices can choose to offload their tasks to be executed in the edge cloud or in local devices. However, in most studies, the computation utilization power of the edge cloud is not considered well. To minimize the energy consumption of mobile devices in MEC system as well as to meet the task completion requirement, we formulate a multi-device task offloading framework for energy optimization based on the computation capacity of the edge cloud. For the future direction, based on this study, we design an efficient strategy to solve these formulated problems.

## ACKNOWLEDGMENT

This research was funded by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology

Research Center) support program (IITP-2021-2017-0-01633) supervised by the IITP (Institute for Information and Communications Technology Planning & Evaluation).

## REFERENCES

[1] Ali Shakarami, Ali Shahidinejad, & Mostafa Ghobaei-Arani (2021). An autonomous computation offloading strategy in Mobile Edge Computing: A deep learning-based hybrid approach. Journal of Network and Computer Applications, 178, 102974.

[2] Fu, Z.; Ren, K.; Shu, J.; Sun, X.; Huang, F. Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans. Parallel Distrib. Syst. 2016, 27, 2546–2559.

[3] Kumar, K.; Lu, Y.H. Cloud computing for mobile users: Can offloading computation save energy. IEEE Comput. 2010, 43, 51–56.

[4] Milan, P.; Jerome, J.; Valerie, Y.; Sadayuki, A. Mobile-Edge Computing Introductory Technical White Paper. White Paper. Available online: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf (accessed on 14 September 2018).

[5] Zhao, M., & Zhou, K. (2019). Selective Offloading by Exploiting ARIMA-BP for Energy Optimization in Mobile Edge Computing Networks. Algorithms, 12(2).

[6] X. Tao, K. Ota, M. Dong, H. Qi and K. Li, "Performance Guaranteed Computation Offloading for Mobile-Edge Cloud Computing," in IEEE Wireless Communications Letters, vol. 6, no. 6, pp. 774-777, Dec. 2017, doi: 10.1109/LWC.2017.2740927.

[7] D. Huang, P. Wang and D. Niyato, "A Dynamic Offloading Algorithm for Mobile Computing," in IEEE Transactions on Wireless Communications, vol. 11, no. 6, pp. 1991-1995, June 2012, doi: 10.1109/TWC.2012.041912.110912.

[8] Y. Hao, M. Chen, L. Hu, M. S. Hossain and A. Ghoneim, "Energy Efficient Task Caching and Offloading for Mobile Edge Computing," in IEEE Access, vol. 6, pp. 11365-11373, 2018, doi: 10.1109/ACCESS.2018.2805798.

[9] M. Chen, Y. Miao, Y. Hao and K. Hwang, "Narrow Band Internet of Things," in IEEE Access, vol. 5, pp. 20557-20577, 2017, doi: 10.1109/ACCESS.2017.2751586.