Energy-Efficient Task Offloading and Resource Scheduling for Mobile Edge Computing

Hongyan Yu[1], Quyuan Wang[2], and Songtao Guo[2]

[1]*Key Laboratory of Intelligent Information Processing and Control of Universities in Chongqing*
[1]*Chongqing Three Gorges University, Wanzhou, Chongqing, 404100 China*
[2]*College of Electronic and Information Engineering, Southwest University, Chongqing, 400715, China*

***Abstract*—Mobile edge computing is an emerging computing paradigm to augment computational capabilities of mobile devices by offloading computation-intensive tasks from resource-constrained smart mobile device onto edge clouds nearby with potential computation capability. However, in general, edge clouds have limited computation resource and energy. Thus it is critical to achieve high energy efficiency while ensuring satisfactory user experience. In this paper, we first formulate the computation offloading problem for mobile edge computing into the system cost minimization problem by taking into account the completion time and energy. We then transform the optimization problem into a convex problem and propose a distributed algorithm consisting of offloading strategy selection, clock frequency configuration, transmission power allocation and channel rate scheduling. Finally, the experimental results show that our algorithm can achieve energy-efficient offloading performance compared to other existing algorithms.**

***Index Terms*—Mobile edge computing, Energy-efficient, Convex optimization, M/M/n queue model.**

I. Introduction

In the era of mobile computing and Internet of Things, mobile devices have become an essential part of modern life. What we face now is the contradiction between inadequate processing capacity of edge devices and the users' ever-growing needs for better performance. Mobile edge computing (MEC) is such a promising technology to solve the conflicts between the computation-intensive applications and the resource-limited mobile devices by shortening the distance between users and clouds [1]. Although MEC transfers the computations to the edge cloud and utilizes task partitioning and offloading technique to enhance the performance significantly [2], it still faces some challenges for realizing more reasonable MEC system. On the one hand, since there may exist multiple edge clouds nearby mobile users in MEC system, it is essential to make the appropriate offloading selection among the edge clouds as well as local device and central cloud. On the other hand, in such an era of frequent communication, not only the computing resource is shortage, but also the communication resource is limited, such as frequency spectrum. It is critical how to exploit limited communication resource to schedule more tasks. In addition, since there are a large number of tasks in MEC system need to be offloaded under the constraint of limited channel resources, therefore, energy saving in the transmission is also critical for improving system energy efficiency.

The objective of this paper is to propose an energy-efficient offloading strategy to minimize total cost by optimizing offloading selection and resource allocation in mobile edge computing system under the assistance of central cloud. Here, the cost includes energy consumption and processing time. Compared with previous work, this paper has several contributions. *First*, by considering that it is much essential to utilize the central cloud to help dealing with complex tasks and monitoring the entire network,we propose a task offloading and resource scheduling mechanism with three offloading destinations, i.e., local device, edge cloud and central cloud. *Second*, we formulate the energy-efficient offloading problem into a cost minimization problem by considering application completion deadline and processing capability constraints and we propose a distributed algorithm containing strategy selection, clock frequency control and transmission power allocation to solve the problem. *Third*, we introduce the queueing theory to formulate an M/M/n queue model with individual capacity between different windows and give the optimal task offloading rate and queue delay by using convex optimization method. *Fourth*, Experimental results show that compared with other existing algorithms, our task offloading and resource scheduling algorithm can reduce about 30% of the cost.

II. System Model and Problem Formulation

In this section, we first introduce the system model. We assume that the set $\mathcal{J} = \{1, 2, ..., J\}$ denotes computation-intensive tasks. Let M denotes the total number of channels, and all channels are employed in a first-come first-serve (FCFS) manner. As shown in Fig.1, our MEC system consists of some mobile phones, multiple access points or base stations and one central cloud. In our MEC system, the edge cloud includes the fixed edge gateway, e.g. base stations, servers in buildings and powerful processing equipment nearby, which assists mobile devices to offload computing tasks, and some mobile phones with abundant resources. Other mobile phones with limited resources is referred to as clients, which can send their requests and offload their tasks onto central cloud through wireless access points. The clients can also offload their tasks onto edge cloud if their connections are available.

978-1-5386-8367-5/18/$31.00 ©2018 IEEE

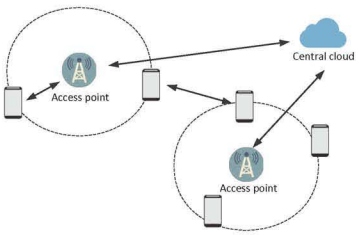


Fig. 1. Illustration of MEC.

A. Local Computing Model

Intuitively, the tasks with smaller size may be executed on local device. Let clock frequency $f_{i,l}$ denote the computation capability of local computing, i.e., processing data size per second. It is known that smart devices can change their clock frequency to deal with different tasks to save energy by dynamic voltage and frequency scaling (DVFS) technique. The execution time of task i by local computing can be expressed as

$$T_{i,l} = \frac{Cy_{i,l}}{f_{i,l}} \tag{1}$$

and the energy consumption is given by

$$E_{i,l} = \kappa Cy_{i,l} f_{i,l}^2 \tag{2}$$

where $Cy_{i,l}$ is the total CPU cycles of computing task i on local device, and κ denotes the effective switched capacitance depending on the chip architecture. In this paper, we set $\kappa = 10^{-11}$ so that the energy consumption is consistent with the measurements in [3]. The expression of local computing cost is derived as follows:

$$Cost_{i,l} = \delta_{i,T} T_{i,l} + \delta_{i,E} E_{i,l} \tag{3}$$

where $0 \leq \delta_{i,T} \leq 1$ and $0 \leq \delta_{i,E} \leq 1$ represent the wights of completion time and energy consumption for task i. In general, we assume that $\delta_{i,T} + \delta_{i,E} = 1$. Clearly, we can control the clock frequency of mobile device to minimize local computing cost.

B. Cloud Computing Model

According to Shannon formula, furthermore, we can compute the data rate for offloading task i as

$$r_i = W log_2(1 + \frac{p_i h_i}{N + \sum_{j \in \mathcal{J} \setminus \{i\}: \alpha_i = \alpha_j} p_j h_j}) \tag{4}$$

where p_i is the transmission power of mobile device offloading task i, and h_i denotes the channel gain between mobile device and cloud. N denotes the thermal noise power and the remaining part of the denominator represents mutual interference power between users. W represents channel bandwidth. As aforementioned, $f_{i,e}$ and $f_{i,t}$ denote the computation capability of edge cloud and central cloud, respectively. Thus the completion time and transmission energy consumption at edge cloud can be computed by

$$T_{i,e} = \frac{d_{i,e}}{r_i} + \frac{Cy_{i,e}}{f_{i,e}} + T_{i,q}^m, \tag{5}$$

$$E_{i,e} = (\frac{d_{i,e}}{r_i} + T_{i,q}^m) p_i + \kappa Cy_{i,e} f_{i,e}^2, \tag{6}$$

respectively. It is known that edge cloud is also resource-restricted. Thus it is necessary to consider the computational energy consumption of edge cloud, which can be calculated by (2). The completion time and transmission energy consumption at central cloud can be given by

$$T_{i,t} = \frac{d_{i,t}}{r_i} + \frac{Cy_{i,t}}{f_{i,t}} + T_{i,q}^m, \tag{7}$$

$$E_{i,t} = (\frac{d_{i,t}}{r_i} + T_{i,q}^m) p_i, \tag{8}$$

respectively, where $T_{i,q}^m$ is the queue waiting time of task i on channel m.

Thus the unified energy-efficiency cost in cloud computing can be given by

$$Cost_{i,c} = \beta_i Cost_{i,e} + (1 - \beta_i) Cost_{i,t} \tag{9}$$

where β_i can be regarded as *Cloud selection factor*, which is used to choose edge clouds or central cloud.

C. Problem Formulation

The total energy-efficiency cost at local execution and cloud execution can be given by

$$Cost_i = \alpha_i Cost_{i,c} + (1 - \alpha_i) Cost_{i,l} \tag{10}$$

where α_i denotes *Offloading selection factor* making a decision on local computing or cloud computing. Thus our total energy-efficiency cost minimization problem can be formulated as:

$$min \sum_{i=1}^{J} Cost_i \tag{11}$$

Subject to $(\forall i \in \{1, 2, ..., J\})$

$$\begin{aligned}
C1 &: \sum_{i=1}^{J} \alpha_i [\beta_i T_{i,e} + (1 - \beta_i) T_{i,t}] + (1 - \alpha_i) T_{i,l} < T_{max} \\
C2 &: \sum_{i=1}^{J} \alpha_i [\beta_i E_{i,e} + (1 - \beta_i) E_{i,t}] + (1 - \alpha_i) E_{i,l} < E_{max} \\
C3 &: \alpha_i \in \{0, 1\}, \beta_i \in \{0, 1\} \\
C4 &: f_{i,e} < f_{i,t}
\end{aligned}$$

In order to transform the non-convex problem to the convex one, we relax the selection factors to a real number between zero and one, i.e., $0 \leq \alpha_i \leq 1, 0 \leq \beta_i \leq 1$.

Theorem 1: The optimization problem with constraints (C1-C4) is convex with respect to (w.r.t) the optimization variables α_i, β_i, $f_{i,l}$, p_i and λ_m.

Proof: We should first prove that the objective function is jointly convex w.r.t the optimization variables α_i, β_i, $f_{i,l}$, p_i and λ_m. Then we show the convexity of constraints . Due to space limit, we omit the detailed proof. ■

III. Distributed Algorithm

A. Processing Capability Optimization

The completion time and energy consumption are associated with CPU clock frequency $f_{i,l}$, if we execute the task on local mobile, *i.e.*, $\alpha_i = 0$. Processing capability optimization aims to optimize CPU clock frequency so as to minimize the total cost when we choose local computing. The cost minimization problem can be rewritten as:

$$\min_{f_{i,l}} \sum_{i=1}^{J} \delta_{i,T} \frac{Cy_{i,l}}{f_{i,l}} + \delta_{i,E} Cy_{i,l} \kappa f_{i,l}^2 \tag{12}$$

$$s.t. \quad \sum_{i=1}^{J} f_{i,l} < J \cdot F \tag{13}$$

where F is the minimum capability of edge clouds. We can find the optimization problem is convex, thus we can employ the convex method [4] to solve this problem.

The optimal CPU clock frequency can be given by

$$f_{i,l}^* = u + v - \frac{1}{3} w_1 \tag{14}$$

where $u = \sqrt[3]{\frac{-b+\sqrt{\Delta}}{2}}, v = \sqrt[3]{\frac{-b-\sqrt{\Delta}}{2}}, a = -\frac{1}{3} w_1^2, b = \frac{2}{27} w_1^3 + w_3, \Delta = b^2 + \frac{4a^3}{27}, w_1 = \frac{\theta}{2\delta_{i,E} Cy_{i,l} \kappa}, w_2 = 0, w_3 = -\frac{\delta_{i,T}}{2\delta_{i,E}\kappa}$.

B. Transmission Power Allocation

If a task is offloaded onto cloud, the offloading rate r_i is determined by transmission power p_i. Thus the cost minimization problem can be transformed into the following transmission power allocation problem:

$$\min_{p_i} \sum_{i=1}^{J} \delta_{i,T} \frac{d_{i,c}}{r_i} + p_i (\delta_{i,E} \frac{d_{i,c}}{r_i} + \delta_{i,T} T_{i,q}^m) \tag{15}$$

It is clear that the optimization problem (15) is convex about transmission power p_i. And according to KKT conditions [4], it is not difficult to observe that the optimal transmission power is the intersection point between two equations, $g(p_i)$ and $G(p_i)$, which can be given by

$$g(p_i) = \frac{\delta_{i,E} \frac{p_i h_i}{N} + \delta_{i,T} \frac{h_i}{N}}{1 + \frac{p_i h_i}{N}},$$

and

$$G(p_i) = \delta_{i,E} ln(1 + \frac{p_i h_i}{N}) - \frac{\delta_{i,T} T_{i,q}^m}{ln2} [ln(1 + \frac{p_i h_i}{N})]^2.$$

Thus we have to utilize the Newton method to achieve its approximate solution, i.e., the transmission power is updated iteratively by

$$p_i(k+1) = p_i(k) - \frac{g(p_i(k)) - G(p_i(k))}{g'(p_i(k)) - G'(p_i(k))} \tag{16}$$

where $g'(p_i)$ and $G'(p_i)$ denote the first-order derivative of $g(p_i)$ and $G(p_i)$ with regard to p_i, respectively.

C. Queue Delay Optimization

We are dedicated to find a set of optimal data offloading rate $\vec{\lambda}^*$ to minimize the queue delay. Thus we can formulate queue delay minimization problem as

$$\min_{\lambda_i^m} \sum_{i=1}^{J} \sum_{m=1}^{M} \alpha_i T_{i,q}^m \quad m \in \{1, 2, ..., M\} \tag{17}$$

where $T_{i,q}^m$ is the queueing delay at channel m, which is given by (7). For channel m, we use a M/M/n queue model of multi-service windows with different capacity to analyze the queue delay.

We can use the state balance equations to get a set of global balance equations, and then we can obtain the queueing delay by simple mathematical operations.

$$T_{i,q}^m = \frac{(1+\gamma)^2 (\rho_i^m)^3 + q(\rho_i^m)^2}{-(1+\gamma^2)(\rho_i^m)^3 + (q-\gamma)(\rho_i^m)^2 + q(\rho_i^m) + \gamma} \frac{1}{\mu} \tag{18}$$

where $\gamma = \frac{\mu_2}{\mu_1}, \rho_i^m = \frac{\lambda_i^m}{\mu}, q = 1 - \varphi + \gamma + \gamma^2 \varphi$.

D. Selection Strategy Optimization

Our first objective function with respect to selection strategy β_i can be written as $Cost_{i,c} = \beta_i Cost_{i,e} + (1 - \beta_i) Cost_{i,t}$, we turn it into the following form:

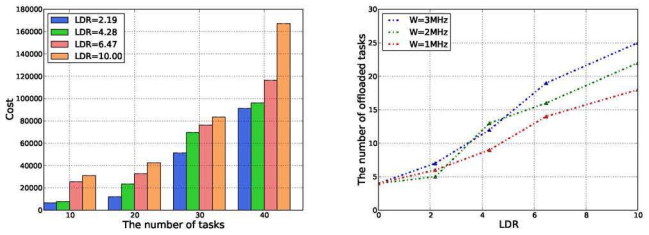
$$\min_{\beta_i} Cost_{i,t} + \beta_i (Cost_{i,e} - Cost_{i,t}) \tag{19}$$

It is clear that the objective function in (19) is a linear function on β_i. We can easily draw the following conclusions. If $Cost_{i,e} > Cost_{i,t}$, we will get the minimum of (19) when we take $\beta_i = 0$. By contrast, if $Cost_{i,e} < Cost_{i,t}$, we will get the minimum of (19) when we take $\beta_i = 1$. Thus, we can obtain the optimal cloud selection. Similar to the computation of β_i, we can compute α_i.

Note that our objective function is divided into four parts to be solved sequentially by using different methods. The time complexity of the proposed algorithm is $\mathcal{O}(J * Iter_{max} * Iter_{p_i})$, where $Iter_{max}$ denotes the maximum number of iterations, and $Iter_{p_i}$ means the number of iterations for transmission power in (16) by Newton iteration method.

IV. Performance Evaluation

In this section, we evaluate the performance for our proposed algorithm and offloading strategy with different specifications. In our simulations, we let link bandwidth $W = 3MHz$ and the noise power be 50 dBm. We set the channel gain $h_i = D^{(-\zeta)}$, where $\zeta = 4$ is the path loss factor and D is the distance between mobile devices and cloud. In order to measure the complexity of the offloading task, similar to [5], we utilize Load-input Data Ratio (LDR), i.e., $LDR = \frac{Cy_i}{d_i}$. If the LDR is high, then the task is complex and vice versa.



(a) (b)

Fig. 2. The impact of task complexity.

A. Impact of Task Complexity

In this subsection, we discuss the impact of task complexity on energy-efficiency cost. We employ LDR to reflect the complexity of tasks. Without loss of generality, we select four values of LDR from 2 to 10 randomly, i.e., let $LDR = 2.19, 4.28, 6.47, 10.00$. Fig.2(a) shows the changes of cost for different LDR and Fig.2(b) depicts the changes of the numbers of offloaded tasks for different LDR.

We can find from Fig. 2(a) that the cost increases rapidly with LDR and the number of tasks. Fig.2(b) shows that as the LDR increases, the number of tasks offloaded onto the cloud increases. According to the definition of LDR, the higher the LDR is, the more complex the task is, thus the more energy it consumes. Fig.2(b) verifies that if a task has high Cy_i and low d_i, then the LDR is high, which means the cost in local computing is higher than that in cloud computing. Thus, if the LDR is low, the task is more suitable for local processing. Otherwise, the task is more suitable for offloading. And Fig.2(b) also describes the relationship between the number of tasks offloaded onto the cloud and the channel bandwidth. Channel bandwidth indicates the condition of communication. When the LDR is fixed, the better communication conditions, the higher the number of offloaded tasks.

B. Comparison of Different Strategies

In this subsection, We compare our strategy with other offloading schemes, i.e., Greedy algorithm [6], Mao's Method in [2] and Dinh's Method in [7] when there are 5 to 30 tasks need to be executed.

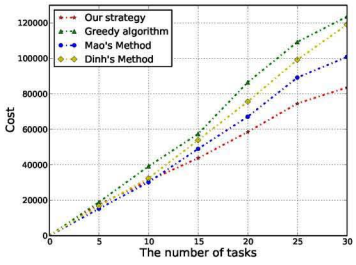


Fig. 3. The comparison of different strategies.

It is not difficult to observe from Fig.3 that the cost by greedy algorithm grows fastest than our scheme. Although Mao's Method and Dinh's Method are slightly better than our method when the number of tasks is small, the advantages of our strategy become more obvious with the number of tasks. This is because greedy algorithm only pursues the optimal value at the current state such that when the number of tasks is larger, its cost grows rapidly. Mao's Method in [2] ignores the central cloud in MEC system, thus lacks the supervision of the central cloud so that the tasks cannot be co-processed by the powerful central cloud in their algorithm. Dinh's Method in [7] mainly focuses on CPU frequency and data rate. However, Dinh's method lacks of scheduling of limited channel resources. Thus a large number of offloaded tasks result in channel congestion and higher communication cost. To improve performance, our designed algorithm not only considers the assistance of the central clod but also optimizes the data delivering rate to avoid channel congestion.

V. Conclusions

In this paper, we propose a task offloading and resource scheduling algorithm to address the minimization problem of total energy consumption and processing time during offloading in MEC system. To the best of our knowledge, this work is the first work on dynamic task offloading and resource scheduling that minimizes total cost by taking into account both the CPU clock frequency control on local device, and the transmission power allocation and the task offloading rate on mobile edge computing with the assistance of the central cloud. Our experimental results show that our task offloading and resource scheduling algorithm outperform other existing approaches in cost-reducing.

VI. Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 61772432), Natural Science Key Foundation of Chongqing (cstc2015jcyjBX0094) and Natural Science Foundation of Chongqing (CSTC2016JCYJA0449).

References

[1] P. Corcoran and S. K. Datta, "Mobile-edge computing and the internet of things for consumers: Extending cloud computing and services to the edge of the network," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 73–74, 2016.

[2] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2017.

[3] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Usenix Conference on Hot Topics in Cloud Computing*, 2010, pp. 4–4.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, 2013.

[5] S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in *IEEE INFOCOM 2016 - the IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[6] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan, "Odessa: Enabling interactive perception applications on mobile devices," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '11. New York, NY, USA: ACM, 2011, pp. 43–56. [Online]. Available: http://doi.acm.org/10.1145/1999995.2000000

[7] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. PP, no. 99, pp. 1–1, 2017.