Computation Offloading Strategy Based on Improved Auction Model in Mobile Edge Computing Network

Liyong Wan¹,²,*
¹School of Software, Jiangxi Normal University, Nanchang, China
²College of Artificial Intelligence, Nanchang Institute of science & Technology, Nanchang, China
*Corresponding author e-mail: wanliyong@jxnu.edu.cn

Abstract. In order to solve the problem of computing power or low energy of users, and to make better use of the idle resources of surrounding mobile devices, the use of D2D direct communication to perform collaborative computing with surrounding mobile devices is an important way to improve resource utilization and reduce local execution energy consumption and delay. To this end, we propose a D2D computing task offloading strategy based on the auction model. Firstly, the requesting device detects the surrounding idle mobile device resources through D2D communication technology, and constructs a communicated topology diagram of the requesting device and the service device. Then, the auction model is used to solve the computing task migration problem of multi-user and multi-service nodes in D2D. At the same time, the offloading problem is converted into an auction model and the task price model is also reasonably defined through comprehensively considering task energy consumption and time delay. The simulation results show that compared with the traditional migration strategy based on communication rate and computing power, the offloading strategy based on the auction model has achieved good results in terms of energy consumption, delay and failure rate.

Keywords: Computation Offloading, Auction Model, D2D, Mobile Edge Computing

1. Introduction
In traditional mobile networks, two mobile devices must communicate with each other through a base station [1]. This mode plays an important role in traditional low-rate services, such as text messages or voice. Tasks offloading are also offloaded to cloud platforms or other servers through the base station [2]. However, with the increase in the number of mobile devices and the enhancement of the computing and storage capabilities of mobile devices, users can search for corresponding devices from the surroundings, and if the other party allows them, they can transfer tasks and can directly use D2D communication technology to directly offload to the surrounding idle and high power mobile phones without base station forwarding, thus it can also reduce the traffic pressure of base station [3]. The offloading strategy is the core issue of computing offloading technology, which focus on strategy problems, such as time overhead and energy consumption, etc. Mao Y and You C [4, 5] studied the
unloading strategy from the energy point of view. Deng M and Tian H [6] researched the offloading strategy in terms of server resources and bandwidth. The literature [7-9] focused on the study of the influence of channel reuse and channel competition on the offloading decision, and does not consider the influence of server resource usage on computing offloading performance. The literature [4, 6, 10] focused on the study of MEC server resources and scheduling, and did not consider the impact of the task's own characteristics (task data size, computing resource requirements, etc.) on computing offloading.

According to the current research status, we design a D2D computation offloading strategy based on the auction model, which is mainly based on D2D communication. The computation offloading strategy comprehensively considers the computation, storage, power and distance, while converts the offloading problem into an auction model, and implements computation offloading through the auction process.

2. Improved D2D Auction Model

2.1. Offloading Decision Model
(1) Offloading Decision. The user needs to perceive and detect the surrounding service nodes that can provide relevant computing resources, and combine the time delay and energy consumption of the task running locally and offloading to the service node to determine whether the task is uninstalled and migrated.

(2) When it is determined that the task needs to be offloaded and executed, the auction algorithm is used, the requester is equivalent to the buyer, and the service node is equivalent to the seller of the service. The two parties decide which service node to perform the offloaded task through the auction process. Only when the service node confirms the execution of the uninstall task and the requesting node also confirms that the task is offloaded to the service node for execution, the transaction is established, which means that the task is successfully scheduled.

2.2. Task Scheduling Based on Improved Auction
When the task is to be run in an offloaded manner, there are two issues to consider in the next step. The first problem is which virtual machine the task should be offloaded to for processing; the second problem is how to schedule tasks when the number of offloading tasks is large and the server computing resources are limited. Aiming at these two problems, we propose a task scheduling strategy based on an improved auction algorithm.

(1) Formalizing the auction model. In the auction model, there are two roles: seller and buyer. The seller provides valuable goods, and the buyer auctions the goods through the valuation of the goods, and the buyer with the higher bid gets the goods. The Auction model of task offloading is Fig.1.

(2) Compensatory auction strategy. In the auction process, consider a special situation: the user buyer uses the price of formula (1) in each auction process to buy, but because the price is lower than
other buyers, the user will not succeed. The goods are bought by auction, and the task cannot be offloaded. Therefore, this pricing principle has a flaw, which is unfairness.

\[ \text{price} = \frac{1}{aE + (1 - a)\tau} \]  

(1)

In response to this problem, we propose a buyer bid compensation mechanism based on the principle of auction. When the buyer fails in the first round of bidding, the price needs to be adjusted in the next bidding, as shown in the price formula (2).

\[ \text{price} = \frac{1}{aE + (1 - a)\tau - \tau X} \]  

(2)

Where \( \tau \) is the length of a time slice, and \( X \) is the number of times the buyer has failed to participate in the auction. The idea of this formula is that if the buyer fails the bidding more often, the buyer's bid price will be higher in the new bid to make the bid successful. Through the analysis of the formula, it is in line with our principle of higher prices as the number of auction failures increases.

2.3. The Offloading Algorithm Based on Improved Auction Model

The algorithm first obtains candidate devices that can be uninstalled for the task of the requesting device, and then evaluates each candidate node according to the evaluation model. In the auction process, the auction is first performed to the service device with the highest bid. Each service device also receives the task with the highest bid. When the task has not been assigned, the task is assigned to the node until all service devices have tasks or tasks have no candidate nodes to assign. Finally, if there are tasks that are not successfully auctioned, the auction will be reevaluated according to the price compensation mechanism until all tasks are allocated to the service equipment.

Algorithm 1. Offloading Auction ( )

Input: The task set \( T \) of requesting device to be offloaded and service device \( S \);

Output: Tasks and the service equipment;

(1) for task, in \( T \):
(2) \( S_i = \text{getService}(\text{task}_i) \)
(3) for s in \( S_i \)
(4) price = \text{Est.}(\text{task}_i, s)
(5) descendent\( (S) \)
(6) for s in \( S \):
(7) \( T = \text{collectTask}(s) \)
(8) task = \text{getMaxPrice}(T)
(9) if task not service:
(10) add task in s
(11) else task = \text{getMaxNextPrice}(T)
(12) collect auction failed task;

3. Simulation Experiment and Result Analysis

3.1. Experiment Parameter Design

In order to verify the effectiveness of the D2D migration strategy, we design a simulation experiment to simulate the migration strategy. The simulation experiment is implemented in java language and the environment is jdk1.8.

We use MATLAB to simulate the D2D offloading strategy experiment based on the auction model. The parameters set in the simulation experiment are shown in Table 1.
Table 1. Simulation Parameter

| Parameter meaning                        | Parameter value range |
|-----------------------------------------|-----------------------|
| Number of mobile terminals              | [5, 20]               |
| Mobile terminal computing power(MIPS)   | [600, 1000]           |
| Mobile terminal communication rate(Mb/s)| [0, 1]                |
| Task calculation amount(MIPS)           | [80, 100]             |
| Task offload size(M)                    | [0, 2]                |
| Mobile terminal transmission energy consumption(J/M) | 0.01 |
| Mobile terminal calculation energy consumption(J/MIPS) | 0.2 |
| Number of relay nodes                   | [2, 5]                |

3.2. Simulation Experiment Results Analysis

In order to verify the effect of the D2D computing offloading strategy based on the auction model, we compare it with the D2D computing task offloading strategy based on the communication rate and the D2D computing task migration strategy based on computing power. In this simulation experiment, we mainly choose three indicators for comparison, which are the average completion time (delay) of the task, the average energy consumption of each task, and the failure rate of the task.

(1) Average Completion Time. The average completion time of a task mainly refers to the average completion time of several randomly generated tasks generated by the system in a simulation. The average energy consumption of a task is also the average energy consumption of completing each task. The average completion time results of the tasks of the three execution strategies are shown in Fig.2.

![Fig.2 Average completion time](image)

The abscissa represents the number of requested devices, and the ordinate is the average completion time of each task. In the experiment, each requesting device will randomly generate 1 to 5 random tasks and set up 10 service devices. In order to ensure the accuracy of the data, each of our data is the average of 10 repeated experiments under the same settings. From the figure, we can see that the migration strategy task based on computing power takes the most time to complete, the migration strategy based on communication rate is the second, and the migration strategy task based on the auction model has the lowest average completion time.

(2) Average Energy Consumption of Tasks. The average task consumption results of the three execution strategies are shown in Fig 3. The abscissa in Fig 3 also shows the number of requested devices, and the abscissa represents the average energy consumption of the task. Each requesting device will generate several random tasks, and the average energy consumption of the experiment results is the average of the energy consumption of 10 repeated experiments under the same settings.
Based on auction model
Based on communication rate
Based on computing power

Fig.3 Average energy consumption

From the results, we can see that the computing power-based migration strategy has the highest energy consumption. Since each task is offloaded, the task needs to be offloaded to the device with the strongest computing power, so the energy consumption is the highest. The migration strategy based on the communication rate selects the service device with the highest communication rate, and the transmission energy consumption is reduced. However, due to weak computing power, the calculation energy consumption is increased. The migration strategy based on the auction model considers energy consumption and time delay, and is better to reduce energy consumption, so the average energy consumption is relatively low.

(3) Task Failure Rate. The results of the average failure rate of tasks for the three execution strategies are shown in Figure 4. In this experiment, the performance of task failure is mainly that the task execution time exceeds the maximum task delay threshold. In the figure, we can see that the failure rate of the migration strategy based on the auction model is low, and the failure rate based on computing power is the highest. And the failure rate based on communication rate is second.

Fig.4 Average failure rate

4. Conclusions
This paper studies the problem of D2D task offloading based on the auction model. D2D task offloading uses D2D communication technology to detect available and idle mobile devices. The D2D can offload tasks to the detected available mobile devices for execution to reduce task processing time delay and local energy consumption. First, establish a communication topology diagram composed of user equipment, relay and forwarding equipment, and service equipment through D2D communication technology. Then estimate the unloading environment based on the task attributes, communication delay, and use the auction method to select the location of the task unloading. Finally, simulation experiments are used to obtain the usability of this strategy for D2D task offloading, which provides a new idea for D2D task offloading.
Acknowledgements
This work was supported by Science and Technology Research Project of Jiangxi Provincial Department of Education under Grant no.GJJ191100.

References
[1] Shi Yusong, Sun Hui, Cao Jie, et al. Edge computing: an emerging computing model for the Internet of everything era[J]. Journal of Computer Research and Development, 2017, 54(5) :907-924.
[2] Satyanarayanan M, Bahl P, Caceres R, et al. The case for VM-based cloudlets in mobile computing [J]. IEEE Pervasive Computing, 2009, 8(4) :14-23.
[3] Jia Mike, Cao Jiannong, Liang Weifà. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks[J]. IEEE Trans on Cloud Computing, 2017, 5(4) :725-737.
[4] Mao Y, Zhang J, Letaief K B. Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices. IEEE Journal on Selected Areas in Communications [J], 2016, 34(12), 3590-3605.
[5] You C, Huang K, Chae H, et al. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading [J]. IEEE Transactions on Wireless Communications, 2017, 16(3), 1397-1411.
[6] Deng M, Tian H, Lyu X. Adaptive sequential offloading game for multi-cell Mobile Edge Computing[C] // 2016 23rd International Conference on Telecommunications (ICT). IEEE, 2016.
[7] Chen X, Jiao L, Li W, et al. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing [J]. IEEE/ACM Transactions on Networking, 2016, 24(5):2795-2808.
[8] Xu D, Li Q. Energy efficient joint scheduling and resource allocation for downlink cognitive radio networks[C]// International Conference on Wireless Communications & Signal Processing. IEEE, 2015.
[9] Lyu X, Tian H, Sengul C, et al. Multiuser Joint Task Offloading and Resource Optimization in Proximate Clouds [J]. IEEE Transactions on Vehicular Technology, 2017, 66(4):1-1.
[10] Yu Bowen, Pu Lingjun, Xie Yuting, et al. Joint task offloading and base station association in mobile edge computing[J]. Journal of Computer Research and Development, 2018, 55(3) :537-550.