



HAL
open science

An Adaptive Delay-Limited Offloading Scheme Based on Multi-round Auction Model for Mobile Edge Computing

Huamei Qi, Yiran Wang, Su Jiang, Chunmeng Yang, Jia Wu

► **To cite this version:**

Huamei Qi, Yiran Wang, Su Jiang, Chunmeng Yang, Jia Wu. An Adaptive Delay-Limited Offloading Scheme Based on Multi-round Auction Model for Mobile Edge Computing. 17th IFIP International Conference on Network and Parallel Computing (NPC), Sep 2020, Zhengzhou, China. pp.310-322, 10.1007/978-3-030-79478-1_27 . hal-03768729

HAL Id: hal-03768729

<https://inria.hal.science/hal-03768729v1>

Submitted on 4 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

An Adaptive Delay-Limited Offloading Scheme Based on Multi-round Auction Model for Mobile Edge Computing*

Huamei Qi¹, Yiran Wang¹, Su Jiang², Chunmeng Yang¹(✉), and Jia Wu¹(✉)

¹ School of Computer Science and Engineering, Central South University, Changsha, People's Republic of China

qhm@csu.edu.cn, 364823753@qq.com, sans_hua@163.com, jiawu5110@163.com

² Information Technology Department, China Life Ecommerce Company Limited Changsha Regional Branch, Changsha Hunan, People's Republic of China
jiangsu8123@163.com

Abstract. Mobile edge computing (MEC) has been widely used in many scenarios due to its advantages such as low latency, high bandwidth, and strong real-time performance. For mobile edge computing in networks with multiple mobile users and multiple MEC servers, this paper proposes an adaptive delay-limited offloading scheme based on a multi-round auction model. First, a multi-round auction model is adopted. Users and servers select each other to achieve a globally optimal match. The proposed evaluation function is based on the delay and energy consumption that can be reduced by offloading, and weights the two parts based on the user's optimization needs. At the same time, a mechanism for adaptive delay limitation is proposed. The threshold is dynamically updated according to the unloading feedback, which makes the algorithm dynamically adapt to the load changes in the network. Experimental results show that the proposed offloading scheme has obvious advantages in reducing the total delay and the total energy consumption of the system, effectively improving the system performance.

Keywords: Mobile edge computing · Computation offloading · Auction model · Delay limit.

1 Introduction

The dramatic increase in the number of mobile terminal devices has driven the development of mobile cloud computing (MCC). However, due to the shortcomings of mobile cloud computing such as high latency [10], mobile edge computing (MEC) came into being [9]. Mobile edge computing has been widely used in fields such as computationally intensive scenarios, intelligent video acceleration, Internet of Vehicles and Internet of Things [7]. Taking a video surveillance system as an example, a mobile cloud computing network captures various video information through a video surveillance device, and then sends it to a cloud surveillance

* Supported by Natural Science Foundation of China (project number: 61803387).

server for processing. This method will transmit huge video data, which not only increases the traffic load of the core network, but also has a high delay. The mobile edge computing network can directly perform data analysis on the MEC server close to the monitoring equipment. This not only reduces the pressure on the network, but also solves the bottleneck problems such as delay and energy consumption of the monitoring system [15].

Some computationally intensive applications consume a lot of resources and energy. Mobile devices have limited resources. This makes computing offloading a key technology for mobile edge computing [23]. Zhang et al. [20] propose optimization of offloading decision based on coordinate descent method in ultra-dense networking. The iterative method finally obtains the optimal offloading decision, which effectively reduces the energy consumption of the system. Wei and Zeng [14] design an edge computing offloading scheme based on Stackelberg game theory. Through the mutual influence and evolution of strategies in a Stackelberg games of "leader with multiple followers", the optimal utility of both parties is achieved. These algorithms have a good effect in a network with one MEC server, but cannot be applied in a network with multiple MEC servers.

There are also some offloading schemes applicable to networks with multiple MEC servers. The computation offloading scheme based on the improved auction model proposed in [11] considers that there are multiple users and multiple servers in the network. The evaluation function of the algorithm only considers the time delay, and does not consider the energy consumption. This is not fair enough for some computing tasks with the goal of reducing energy consumption. Lin [8] uses a multi-round auction algorithm, which is applicable to the situation of multiple users and multiple MEC servers in the network. The effect of actual offloading is not considered in the evaluation function. This standard is not objective enough.

The above solutions have achieved good results, but the following problems still exist.

1) The offloading scheme based on the auction model can realize the bidirectional selection of multiple users and multiple MEC servers in the network. But the evaluation function given in the above schemes all have deficiencies.

2) None of the existing algorithms consider the impact of delay on the network. Some algorithms give users a fixed delay tolerance, but it is difficult to meet the needs of the entire network.

Based on the above analysis, this paper proposes an adaptive delay-limited offloading scheme based on the multi-round auction model. The contributions of this paper are listed as follows.

1) The proposed evaluation function in the multi-round auction model is based on the delay and energy consumption that can be reduced by offloading, and weights the two parts based on the user's optimization needs.

2) A mechanism for adaptive delay limitation is proposed. The threshold is dynamically updated according to the offloading feedback, which makes the algorithm dynamically adapt to the load changes in the network and improve the overall performance of the network.

The rest of this paper is organized as follows. Section 2 concludes the related work. The offloading scheme proposed in this paper is described in Section 3, including four parts: scenario description, calculation model, multi-round auction model, and adaptive delay-limited mechanism. Section 4 demonstrates the simulation results. Finally, section 5 concludes this study.

2 Related work

Offloading schemes with different optimization goals are suitable for different types of computing tasks [5]. Offloading schemes with the goal of reducing latency are suitable for time-sensitive tasks, while offloading schemes with the goal of reducing energy consumption are suitable for energy-constrained tasks. In order to better meet the needs of users, the scheme proposed in this paper belongs to the offloading scheme with the trade-off between latency and energy consumption, because the evaluation function proposed in this paper is related to both latency and energy consumption. According to different optimization goals, the existing offloading schemes in mobile edge computing can be divided into the following three.

2.1 Offloading scheme with the goal of reducing latency

The distributed computing offload algorithm designed in [3] quantifies the calculation delay index to achieve lower calculation time overhead. Wang et al. [12] propose a scheme that considers interference management when offloading, and allocates resources by minimizing delay. Chen et al. [2] use the idea of software-defined networking. The task offloading problem is expressed as a mixed integer nonlinear calculation process, and the problem of reducing delay is transformed into task offloading placement problem and resource allocation sub-problem. Compared with random unloading and unified unloading, this scheme can shorten the delay to a greater extent. Yu et al. [17] propose a complete polynomial time approximation scheme, which effectively shortens the calculation delay. Yuan and Cai [19] transform the optimal content offloading problem into the content maximum delivery rate problem, thereby reducing the delay. The evaluation function proposed in [11] is based on the effect of reducing latency.

2.2 Offloading scheme with the goal of reducing energy consumption

Zhang et al. [21] use an artificial fish school algorithm, and experiments show that the algorithm has a significant reduction in energy consumption. Zhao et al. [24] propose a greedy heuristic method based on linearization and Gini coefficient to minimize system energy consumption. Geng et al. [6] use heuristic search algorithm to solve offload decision and task scheduling problems, which can significantly reduce the energy consumption of mobile devices. Zhang et al. [22] propose an energy-aware offloading scheme, which optimizes the allocation of communication and computing resources through an iterative search algorithm,

thereby reducing the energy consumption of mobile devices. Xu et al. [16] propose a particle swarm optimization algorithm for energy consumption optimization, which converges stably.

2.3 Offloading scheme with the trade-off between latency and energy consumption

Dai et al. [4] propose a calculation offload and resource allocation mechanism based on the multiplier method to solve the convex optimization problem of minimum energy consumption and delay weighted sum. Wang et al. [13] design a computational offload algorithm (CAMEGA) based on improved genetic algorithm with the goal of minimizing the weighted sum of delay and energy consumption. Yu et al. [18] propose a power distribution algorithm based on game theory, and uses a binary search method to optimize transmission power to reduce transmission delay and energy consumption. Zhao [25] designs a heuristic algorithm based on simulated annealing to find the optimal solution. Bozorgchenani et al. [1] model task offloading in MEC as a constrained multi-objective optimization problem (CMOP) that minimizes both the energy consumption and task processing delay of the mobile devices. Lin [8] designs an evaluation function which is the weighted sum of the user's demand for delay and energy consumption.

3 Adaptive Delay-Limited Offloading Scheme Based on Multi-round Auction Model

3.1 Scenario description

The MEC servers are deployed on the small base stations in the network to process users offloaded tasks. They have strong real-time performance and they can provide a good user experience. Suppose that there are n mobile users in the network, forming a set $U = \{U_1, U_2, \dots, U_n\}$, and m MEC servers, forming a set $S = \{S_1, S_2, \dots, S_m\}$.

3.2 Calculation model

Local execution Assume that the required tasks for user U_i is x bits. Each bit of computation requires the CPU to run C cycles. The user's local computing capability is F_{local} as cycle/s. Let P_{local} be the local compute power, the time and energy required for local execution can be defined as below:

$$T_{local} = \frac{xC}{F_{local}} \quad (1)$$

$$E_{local} = \frac{xC}{F_{local}} \times P_{local} \quad (2)$$

Where T_{local} is the time required for local execution, and E_{local} is the energy consumption required for local execution.

Offloading execution Assume that the user U_i offloads the computing task to the MEC server S_j . The bandwidth allocated to each base station is B . The channel noise power is σ^2 . The channel gain is H_n . The transmit power is P_{send} . The upload rate R_{send} can be obtained:

$$R_{send} = \frac{B}{n} \times \log_2\left(1 + \frac{P_{send} \times H_n}{\sigma^2}\right) \quad (3)$$

Similarly, the download rate $R_{receive}$ from the MEC server to the mobile user can be obtained.

The offloading time consists of three parts: task upload time, task calculation time on the server, and calculation result download time.

$$T_{offload} = T_{calculate} + T_{send} + T_{receive} \quad (4)$$

$$T_{calculate} = \frac{xC}{F_{server}} \quad (5)$$

$$T_{send} = \frac{x}{R_{send}} \quad (6)$$

$$T_{receive} = \frac{\alpha x}{R_{receive}} \quad (7)$$

Where $T_{calculate}$ is the task compute time on the server. F_{server} is the server compute capability as cycle/s. T_{send} is the task upload time. $T_{receive}$ is the result download time. α is the data compression ratio between the calculated result and the original data.

As calculated results are compressed, the energy consumption of download is negligible. We then get:

$$T_{offload} = \frac{xC}{F_{server}} + \frac{x}{\frac{B}{n} \times \log_2\left(1 + \frac{P_{send} \times H_n}{\sigma^2}\right)} + \frac{\alpha x}{\frac{B}{n} \times \log_2\left(1 + \frac{P_{receive} \times H_n}{\sigma^2}\right)} \quad (8)$$

$$E_{offload} = \frac{xP_{send}}{R_{send}} \quad (9)$$

Where $T_{offload}$ is the offloading time. $E_{offload}$ is the offloading energy consumption.

Offloading conditions The computing resources required for the offloading tasks cannot exceed the total resources provided by the MEC servers. At the same time, The tasks that can be offloaded need to meet the following conditions:

$$T_{offload} \leq T_{local} \quad (10)$$

$$E_{offload} \leq E_{local} \quad (11)$$

3.3 Multi-round auction model

The multi-round auction model consists of evaluation stage and auction stage. According to formulas (8) and (9), it can be seen that due to the differences in channel conditions and the servers, the time and energy required for the same task to be offloaded to different MEC servers are various. The multi-round auction model allows users and MEC servers to perform a two-way selection, which is suitable within multiple users and MEC servers scenarios. More details of the multi-round auction model refers to [8].

Evaluation stage In the user's evaluation stage, each user performs a separate evaluation on all cases of offloading to different MEC servers, and the improved evaluation function proposed in this paper is:

$$price = \omega_t \times \rho_t \times (T_{local} - T_{offload}) + \omega_e \times \rho_e \times (E_{local} - E_{offload}) \quad (12)$$

$$\omega_t + \omega_e = 1 \quad (13)$$

Where ρ_t and ρ_e represent the bid constants of delay and energy consumption. ω_t and ω_e are the weights of delay and energy consumption. The users determine ω_t and ω_e according to the task type and different needs. For example, $\omega_t = 1$ and $\omega_e = 0$, which indicates only low delay is expected for the task offloading and not energy consumption.

After the users evaluate the resources of each MEC server, they give each element in the respective set $S = \{S_1, S_2, \dots, S_m\}$ a price. $S_j.price$ represents the evaluation of the user offloading the task to the S_j server. Find the element with the biggest value of $S_j.price$ in the set S , S_j is determined to be the highest priority MEC server corresponding to the task, and $S_j.price$ is the bidding price for S_j of the task. As the larger the value of $price$, the better the MEC server fulfillment of users need.

Auction stage In the auction stage, each MEC server allocates resources to tasks which select this server as the highest priority server. The order of resource allocation is determined by the bidding prices of these tasks. After the first round of auction, if some tasks fail to be offloaded and there are remaining server resources, the second round of auction will be conducted. The S set will be updated, and the task will participate in the auction again with the new bid until no resources remain or the task is completely offloaded.

3.4 Adaptive delay-limited mechanism

Based on the multi-round auction model, this paper proposes an Adaptive delay-limited mechanism. Since the success of the auction depends on the $price$, the time for the task to wait for the final auction is mainly related to the $price$. The smaller the value of $price$ is, the more likely the auction fails, and the longer the waiting time will be.

Algorithm 1 The Offloading Algorithm**Input:** The number of MEC servers m , the number of users n , and the parameter b .**Output:** The match matrix $match$.

```

 $price \leftarrow \rho_t * \omega_t * T_{dis} + \rho_e * \omega_e * E_{dis}$ 
 $P_0 \leftarrow (1 - b) * P_{max}$ 
// Find tasks performed locally
 $[row, cell] \leftarrow find(price < P_0)$ 
 $price(:, cell) \leftarrow 0$ 
 $j \leftarrow length(cell)$ 
while  $j < n$  do
  // Record the  $price$  of tasks with big  $T_{wait}$ 
   $[row, cell] \leftarrow find((T_{local} - T_{offload} < T_{wait}))$ 
   $P(1, cell) \leftarrow price(1, cell)$ 
   $temp\_p \leftarrow price$ 
  // Match the best task for each server
  for  $i \leftarrow 1$  to  $m$  do
     $max\_p \leftarrow max(max(price))$ 
     $[row, cell] \leftarrow find(temp\_p \equiv max\_p)$ 
     $temp\_p(:, cell) \leftarrow 0$ 
     $temp\_p(row, :) \leftarrow 0$ 
     $price(:, cell) \leftarrow 0$ 
     $T_s(1, cell) \leftarrow T_{calculate}(1, cell)$ 
     $match(row, k) \leftarrow max\_p$ 
    // Task offloaded successfully
     $j++$ 
  end for
   $T_{wait} \leftarrow T_{wait} + max(T_s)$ 
   $k++$ 
end while
 $P_{max} \leftarrow max(P_{max}, max(P))$ 

```

If there are many tasks in the network, some of them will fail to be offloaded. Since the server computing capacity has a greater advantage than the local computing capacity, the time waiting for the next auction has little effect on tasks with not very strict latency requirements. The optimization effect brought by persistent offloading is still considerable. However, for tasks with small values of $(T_{local} - T_{offload})$ and strict latency requirements, their $price$ is relatively small. These tasks are likely to fail in multiple auctions. The delay caused by waiting for the next auction may have a great impact on them. If these tasks exit the auction early and choose to execute locally, the result can better meet their needs. At the same time, the latency of other tasks in the network will be reduced, and the burden on the entire network will be reduced.

Algorithm 1 shows the offloading algorithm proposed in this paper. The algorithm is comprising of threshold update and condition verification. A nested loop is used to match user and server resources.

The $price$ of all tasks with a queue delay longer than $(T_{local} - T_{offload})$ are recorded in the set P . Find the maximum value in P and assign it to P_{max} .

Set a threshold P_0 , which needs to be updated according to the change of the network. After each time period, update P and P_{max} to get a new P_0 :

$$P_{\max} = \max\{P_{\max}, \max P\} \quad (14)$$

$$P_0 = (1 - b)P_{\max} \quad (15)$$

Where $b \in (0, 1)$. Its role is to provide a detection range for the estimation of P_0 , allowing the threshold to change up and down.

Before each auction, an additional verification is required. The tasks to participate in the auction must meet the following conditions:

$$price \geq P_0 \quad (16)$$

It can be seen that when the network load is heavy, additional conditions will become stricter to maximize the overall benefit. When the network load is light, the probability of failure in auctions will be greatly reduced, and the value of P_0 will also adapt to changes in the network and become very small, which will allow more computing tasks to participate in the auction.

4 Simulation results and analysis

The experiment assumes that there are three MEC servers in the network, the number of users increases from 100 to 1000, and each user requests to offload a task. Other parameters in the experiment are shown in Table 1.

Table 1: The simulation parameters

Parameter	Value
Bandwidth allocated to each base station	20MHz
Channel noise	-170 dBm/Hz
Channel gain	[-50,-30] dBm
CPU operating capacity	1000 cycles/bit
Computing capability of local CPU	[0.1,1] GHz
Computing capability of MEC server	100 GHz
Calculation amount of each task	[50,200] kB
Local computing power	0.1W
Local transmit power	0.1W
Base station transmit power	1W
Data compression rate	0.2

Fig.1 compares the delay and energy consumption of tasks executed locally and offloaded in different ways. The experiment was conducted under the condition of different number of users. The delay and energy consumption of all users were accumulated to obtain the total delay and total energy consumption. The evaluation function in [8] is only based on the needs of users, and the offloading effect of this scheme is difficult to compare with other schemes. So the offloading scheme in [11] was used for this experiment, which compensates tasks with

T_{wait} through the evaluation function. It can be seen that the multi-round auction scheme with adaptive delay limitation proposed in this paper has certain advantages in reducing delay and energy consumption. The reason for the advantage is that the optimization objective of the evaluation function contains a reduction in delay and energy consumption. Another reason is that the multi-round auction ensures that each user and each MEC server can get the best global match in each auction .

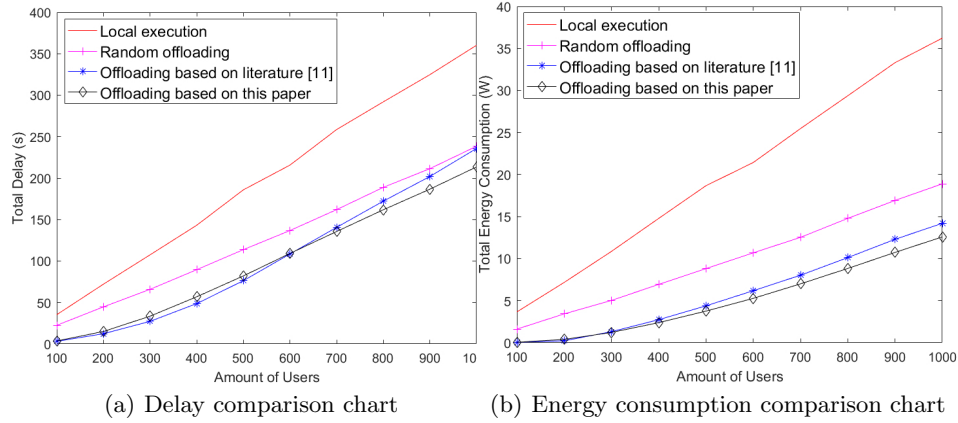


Fig. 1: Comparison chart of different offloading methods

Fig.2 is a delay comparison chart of different delay limiting methods. The experiment was conducted on the basis of using the same multi-round auction model. For the offloading method with fixed delay limit, after experimental comparison, the best performance was achieved in all aspects when T_0 was 0.3s. It can be seen that when the number of users is small, the total delay of the adaptive-delay-limited method and the non-delay-limited method is significantly lower than that of the fixed-delay-limited method. This is because some tasks can only be executed locally because they do not meet the fixed delay limit, which increases the delay. When the number of users is large, the total delay of the offloading method with fixed delay is the lowest. Because of the fixed delay limit, Some users no longer insist on offloading, which reduces the queuing delay and reduces the total delay. Although the total delay of the adaptive-delay-limited method is not the lowest, the advantages are also obvious compared with the non-delay-limited method.

Fig.3 is an energy consumption comparison chart of different delay limiting methods. It can be seen from the figure that the total energy consumption of the non-delay-limited method is always the lowest of the three. All tasks are offloaded, and users only need to consume the energy when uploading. However, in practical applications, the non-delay-limited method is hard to be accepted, because the continuous queuing delay will make the total delay increase significantly. The fixed-delay-limited method has the highest total energy consumption. This is because the fixed delay limit causes many tasks to be executed locally,

which increases user energy consumption. The adaptive-delay-limited method is significantly better than the fixed-delay-limited method in terms of energy consumption.

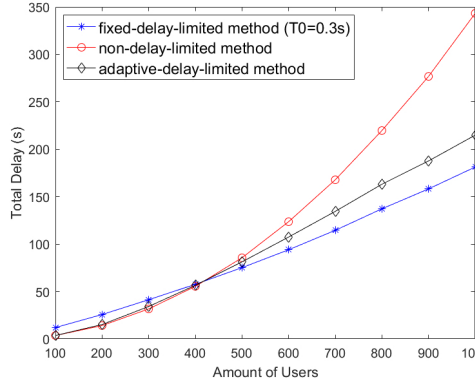


Fig. 2: Delay comparison chart of different delay limiting methods

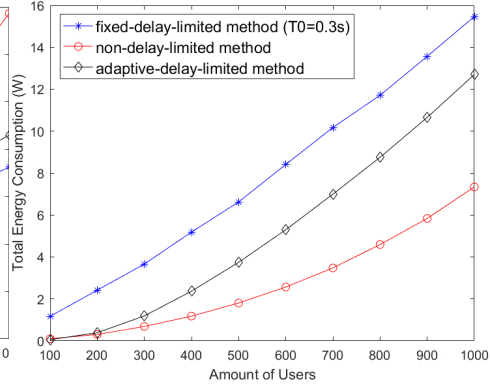


Fig. 3: Energy consumption comparison chart of different delay limiting methods

Combining the delay result and energy consumption result of offloading, it can be known that the offloading method with adaptive delay limitation has obvious advantages. This is because the offloading effect caused by the load change will give the mechanism a feedback, so that P_0 is constantly updated. The delay limit can find the best value in a short time to adapt to the load changes in the network.

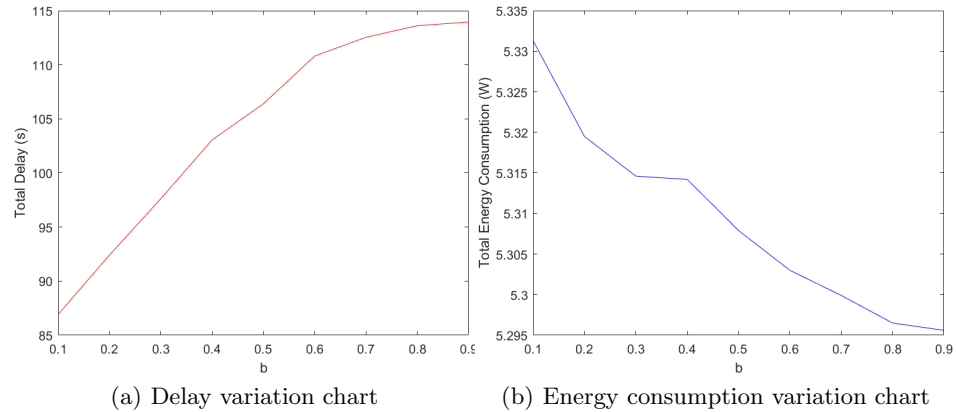


Fig. 4: Variation chart of different b

In addition, the experiment proves that when the number of users is fixed, different values of b will bring different offloading effects. Fig.4 shows the variation of delay and energy consumption with different value of b . The larger the value of b , the higher the total delay, and the lower the total energy consumption. This is because a larger b means that the delay limit is more relaxed. The more

tasks that will be successfully offloaded, the more energy consumption will be reduced. But the queuing delay causes the delay advantage to be weakened. In other words, the smaller the value of b , the more obvious the delay reduction. The larger the value of b , the more obvious the reduction in energy consumption. Therefore, the multi-round auction offloading scheme with adaptive delay limitation proposed in this paper can be applied to networks with different densities, and can also be applied to networks with different emphasis on optimization goals through the adjustment of b .

5 Conclusion

In the research of mobile edge computing offloading, in order to make the offloading scheme better applicable to the network of multiple users and multiple MEC servers, this paper adopts a multi-round auction model, in which users and MEC servers select each other to achieve the global best match. Based on the time delay and energy consumption actually reduced by offloading, and considering the optimization needs of users, a better evaluation function is proposed. This paper also designs an adaptive delay limit mechanism to continuously adapt to load changes and find the optimal delay limit to improve the overall performance of the network. Experiments show that the offloading scheme proposed in this paper can effectively reduce the total delay and total energy consumption of network devices in networks with different loads. And compared with the fixed delay limited method and non-delay limited method, it has obvious advantages.

References

1. Bozorgchenani, A., Mashhadi, F., Tarchi, D., Monroy, S.S.: Multi-objective computation sharing in energy and delay constrained mobile edge computing environments. *IEEE Transactions on Mobile Computing* pp. 1–1 (2020)
2. Chen, M., Hao, Y.: Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE Journal on Selected Areas in Communications* **36**(3), 587–597 (2018)
3. Chen, X., Jiao, L., Li, W., Fu, X.: Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking* **24**(5), 2795–2808 (2015)
4. Dai, M., Liu, Z., Guo, S., Shao, S., Qiu, X.: Edge computing offload and resource allocation mechanism based on terminal energy consumption and system delay minimization. *Electronics and Information Science* **41**(11), 2684–2690 (2019)
5. Dong, S., Li, H., Qu, Y., Zhang, Z., Hu, L.: Review of researches on computing offloading strategies in mobile edge computing. *Computer Science* **46**(11), 32–40 (2019)
6. Geng, Y., Yang, Y., Cao, G.: Energy-efficient computation offloading for multicore-based mobile devices. In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. pp. 46–54. IEEE (2018)
7. Li, Z., Xie, R., Sun, L., Huang, T., et al.: Overview of mobile edge computing. *Telecommunication Science* **34**(1), 87–101 (2018)

8. Lin, X.: Calculation task offloading strategy based on resource joint configuration in mobile edge computing network. Master's thesis, Beijing University of Posts and Telecommunications (2017)
9. Mach, P., Becvar, Z.: Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials* **19**(3), 1628–1656 (2017)
10. Pan, J., McElhannon, J.: Future edge cloud and edge computing for internet of things applications. *IEEE Internet of Things Journal* **5**(1), 439–449 (2017)
11. Sheng, J., Teng, X., Li, W., Wang, B.: Calculation offloading strategy based on improved auction model in mobile edge computing. *Computer Application Research* (2019/04/26)
12. Wang, C., Yu, F.R., Liang, C., Chen, Q., Tang, L.: Joint computation offloading and interference management in wireless cellular networks with mobile edge computing. *IEEE Transactions on Vehicular Technology* **66**(8), 7432–7445 (2017)
13. Wang, Y., Ge, H., Feng, A.: Computing offloading strategy for cloud-assisted mobile edge computing. *Computer Engineering* (2019/10/25)
14. Wei, Z., Zeng, L.: Edge computing offloading decision-making method based on stackelberg game theory. *Mathematics Practice and Understanding* **49**(11), 91–100 (2019)
15. Xie, R., Lian, X., Jia, Q., Huang, T., Liu, Y.: Summary of mobile edge computing offloading technology. *Journal of Communications* **39**(11), 138–155 (2018)
16. Xu, J., Li, X., Ding, R., Liu, X.: Multi-resource calculation offloading strategy optimized for energy consumption in mobile edge computing. *Computer Integrated Manufacturing System* **25**(4), 954–961 (2019)
17. Yu, R., Xue, G., Zhang, X.: Application provisioning in fog computing-enabled internet-of-things: A network perspective. In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. pp. 783–791. IEEE (2018)
18. Yu, X., Shi, X., Liu, Y.: Joint optimization of offloading strategy and power in mobile edge computing. *Computer Engineering* (2019-12-02)
19. Yuan, P., Cai, Y.: A greedy strategy for content offloading in mobile edge computing. *Computer Application* **39**(9), 2664–2668 (2019)
20. Zhang, H., Li, H., Chen, S., He, X.: Task offloading and resource optimization based on mobile edge computing in ultra-dense networks. *Electronics and Information Science* **41**(5), 1194–1201 (2019)
21. Zhang, H., Guo, J., Yang, L., Li, X., Ji, H.: Computation offloading considering fronthaul and backhaul in small-cell networks integrated with mec. In: *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. pp. 115–120. IEEE (2017)
22. Zhang, J., Hu, X., Ning, Z., Ngai, E.C.H., Zhou, L., Wei, J., Cheng, J., Hu, B.: Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks. *IEEE Internet of Things Journal* **5**(4), 2633–2645 (2017)
23. Zhang, K., Gui, X., Ren, D., Li, J., Wu, J., Ren, D.: Review of research on computing migration and content caching in mobile edge networks. *Journal of Software* **30**(8), 2491–2516 (2019)
24. Zhao, P., Tian, H., Qin, C., Nie, G.: Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing. *IEEE Access* **5**, 11255–11268 (2017)
25. Zhao, Y.: Research on computation offloading in mobile edge computing systems with limited resources. Master's thesis, Beijing University of Posts and Telecommunications (2019)