An Optimal Choice for QoE with Cooperative Mobile-edge-computing in UDN

Qi Li and Juanjuan Sun

ABSTRACT

MEC and UDN are two key technologies in the field of wireless communication. In a UDN network, deploying MEC servers in both macro and micro base stations can effectively improve user experience. However, according to the actual scenario, the processing capability of each MEC server is different. If the collaborative processing of the server is not considered, some servers may be busy and idle, and how to design a server collaborative processing strategy to get the best user experience is an issue that needs to be addressed. Therefore, we first establish a system model of the topological MEC server under UDN, and then establish the objective function considering delay and energy consumption. In order to find the cooperation strategy between servers, we use cooperation algorithm to find the optimal solution. Finally, the results are compared with the baseline algorithm. The results show that the collaborative strategy can perform better time and energy consumption.

KEYWORDS

MEC, UDN, Cooperation policy, User experience.

INTRODUCTION

At present, with the commercialization of 5G, many technologies are gradually maturing. In order to meet the conditions of high-capacity, low-latency, high-reliability and other user experience enhancements, the researchers proposed a new network architecture: Ultra-Dense Networks (UDN), which increases the peak rate and flow density of data flow by increasing the deployment density of base stations[1]. To meet the demands of plenty of smart mobile devices, Mobile Edge Computing (MEC) has been proposed by researchers.

European Telecommunications Standards Institute (ETSI) defines MEC as providing IT services and cloud computing capabilities at the mobile edge[1]. A simple understanding is to run a cloud server at the edge of the mobile network. The MEC server can be sunk to the Radio Access Network (RAN), that is to say, Calculate and store closer to the base station location.[2] Distribute some traffic to the MEC server of the near user, instead of going to the remote Core Network (CN), and this can efficiently reduce the time-delay. Besides, one of the networking modes of the ultra-dense network is the deployment mode of the macro base station plus the micro base station. The introduction of micro base stations can significantly reduce energy consumption. Time-delay and energy con-
sumption are two aspects that need to be considered in the user-centric scenario to improve the user experience.

There are many research points about MEC. [3] proposes a new Joint Offloading and Resource Allocation Scheme for Mobile Edge Computing. Considering the actual deployment situation, the server for mobile edge computing may be heterogeneous. Paper [4] proposes an optimization strategy considering the computational offload strategy in this scenario. In [5], the author adopted a distributed game approach to balance the cost and latency tradeoff. The paper [6] captures a Quality-of-Experience (QoE) based cost function, as we know, QoE's evaluation subject is the user, so the author introduced a weighing factor $\omega_{QoE}$ to represent the user preference for latency and energy consumption. [7] and [8] concentrate on the access selection and resource management in Ultra-Dense Networks with Mobile Edge Computing. In general, there are three types of edge calculations that currently appear: mobile edge computing, micro cloud computing and fog computing. The authors propose a joint QoE function with fairness and design a Fairness Cooperation Algorithm to obtain the optimal fairness cooperation policy for all fog nodes in fog computing system [9]. Inspired by these papers, we construct a cooperative mobile edge computing system in UDN, our purpose is to find the optimal choice to satisfy QoE cost function. In addition, we calculated the average processing time and energy consumption for this system, and simulated the resulting graphs and compared with the algorithm without the cooperative strategy.

The contribution of our work can be concluded as follows: First, we consider a scenario that in UDN system, both the macro base station and the micro base station are topologically connected to the MEC server. Second, we consider cooperation of MEC Server in our problem to encourage the owner of the MEC servers - base stations taking part in collaboration. And then, we propose the optimization problem as minimizing joint response time and energy. Finally, we propose Genetic algorithm to solve this optimization problem, and compare the simulation results of this algorithm and the unused cooperative strategy algorithm. The results show that, compared to no cooperation strategy, the MEC server adopting a cooperative strategy can effectively reduce response time and energy consumption.

The rest of this paper is organized as follows. In Section II, we formulate the cooperative mobile-edge-computing in Ultra-Dense network in detail. In Section III, we propose a cooperation algorithm for the optimal cooperation choice. We implement our algorithm in a simulation environment and perform the simulation results in order to find the optimal choice in Section IV. Section V draws the conclusion and provides future directions.

**SYSTEM MODEL**

In this section, we first introduce the UDN system model with MEC computing. Then we describe overall cooperation offloading process among all base station in the UDN system and formulate them. Finally, we discuss the Objective function and get the equation of optimization problem.
SYSTEM MODEL

A single macro base station located in the center and several micro base stations randomly scattered in the field, all of them are equipped with MEC servers. The UEs dispersed in the cell. A diagrammatic sketch of the MEC enhanced UDN is shown in Figure 1.

In this system, there are two options for the amount of tasks from the user to each base station, processing at the local base station or offloading to the other base stations for processing. For ultra-dense networks, offloading to other base stations including offloading to the same type of micro base station and offloading to the macro base station.

In this paper, we mainly focus on the cooperation mechanism of processing tasks among base stations, so in this problem, we do not consider the uplink, downlink time and energy consumption between the terminal and the base station. Assume that each terminal has connected to a nearby micro base station according to the maximum SINR principle, and uploads tasks totally.

![Figure 1. System Model with MEC.](image)

In the scene shown in Figure 1, let the number of micro base stations is N, labeled as set $\mathcal{I}$, and also a macro station, so there are a total of N+1 base stations. The base stations are heterogeneous, that is to say, the processing power of the MEC servers mounted by each base station is different. The processing power of the macro base station is several times that of the micro base station. The macro base station does not accept the direct connection of users and only processes the workload of other micro base stations to be unloaded. The amount of tasks to reach each micro base station $i$ is $\lambda_i$ and $\Lambda = \langle \lambda_i \rangle$, $i \in \mathbb{N}$.

We divide the task amount of the i-th base station into N+1 parts, the 1to N parts shares are processed by the same kind of micro base station, the N+1th part is processed by the macro base station, and we use $k$ to indicate the proportion of allocation, so $k_i$ is the proportion of the i-th base station's task processed locally, $k_{ij}$ is the ratio assigned to other base stations, in particular, $k_{iN+1}$ is the ratio assigned to the macro base station. Considering two extreme situations, for the i-th base station’s task, all the tasks are processed locally or totally by other base stations, we get:

$0 \leq k_{ij} \leq 1, \ i \in \mathcal{I}, j \in \mathbb{N} + 1$, in the same way, $0 \leq k_{iN+1} \leq 1$. So we can get a matrix of N rows of N+1 columns, denoted as $K$, signaled as follows:

<table>
<thead>
<tr>
<th>$k_i$</th>
<th>$k_{i1}$</th>
<th>$k_{i2}$</th>
<th>$\cdots$</th>
<th>$k_{iN}$</th>
<th>$k_{iN+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>$\lambda_2$</td>
<td>$\lambda_3$</td>
<td>$\cdots$</td>
<td>$\lambda_N$</td>
<td>$\lambda_{N+1}$</td>
</tr>
</tbody>
</table>

448
As mentioned in the previous section, we do not consider the uplink, downlink time and energy consumption between the terminal and the base station. Therefore, for the i-th base station, the process to deal with the tasks considering collaborating with each nearby base station can be divided into three parts:

- \( k_{ii} \times \lambda_i \) amount of tasks is handled by the i-th base station itself, i.e. part I;
- \( \sum_{j \neq i} k_{ij} \times \lambda_i \) amount of tasks is transmitted to other base stations, i.e. part II; \( i \in N, j \in \{1,2,\ldots,N+1\} \);
- \( \sum_{j \neq i} k_{ij} \times \lambda_i \) amount of tasks is handled by other base stations, i.e. part III; \( i \in N, j \in \{1,2,\ldots,N+1\} \).

Each part has time delays and energy consumption, the time delay and energy consumption of each part are respectively \( t_1, t_2, t_3, e_1, e_2, e_3 \). The calculation formulas for time and energy consumption will be given separately below.

**EQUATIONS**

At each base station, we consider an M/M/1 queueing system to deal with the tasks in each time unit. In part I, we can get the time-cost and energy consumption as follows:

\[
t_1 = \frac{1}{b_i - k_{ii} \times \lambda_i} \quad (2)
\]

\[
e_1 = p_i \times \frac{1}{b_i - k_{ii} \times \lambda_i} \quad (3)
\]

Similarly, we can get the time-cost and energy consumption in part III, as follows:

\[
t_3 = \frac{1}{b_j - \sum_{j=1}^{N+1, j \neq i} k_{ij} \times \lambda_i} \quad (4)
\]

\[
e_3 = p_i \times \frac{1}{b_j - \sum_{j=1}^{N+1, j \neq i} k_{ij} \times \lambda_i} \quad (5)
\]
In order to reduce the computational complexity, we assume that the power of all base stations processing the task is a constant, donated as \( p_t \) and the transmission power is a constant, donated as \( p_r \).

Let’s consider part II now, according to Shannon formula, transmission rate are related to bandwidth and signal-noise-ratio, the transmission rate from base station \( i \) to \( j \) must be able to support the workload, which is \( k_{ij} \cdot \lambda_i \), because our ultimate goal is to minimize time and energy consumption, let the transmission rate equal this expression, so we get:

\[
t_2 = \sum_{j=1}^{N+1} \frac{1}{k_{ij} \cdot \lambda_i}
\]

\[
e_2 = \sum_{j=1}^{N+1} p_r \cdot \frac{1}{k_{ij} \cdot \lambda_i}
\]

Thus, for base station \( i \), its time cost and energy consumption are recorded as \( T_i \) and \( E_i \) respectively, so we have:

\[
T_i = t_1 + t_2 + t_3
\]

\[
E_i = e_1 + e_2 + e_3
\]

Thus, the total time cost and energy consumption:

\[
T = \sum_{i=1}^{N} T_i
\]

\[
E = \sum_{i=1}^{N} E_i
\]

OBJECTIVE FUNCTION.

Taking into account the time delay and energy consumption, the objective function is set to the QoE function. Considering that each user’s preferences are different, some users may prioritize shorter delays and not pay attention to the consumption of power. Some users may have the opposite, so a weighting factor \( \beta_{QoE} \) is introduced in this objective function. We defined \( 0 \leq \beta_{QoE} \leq 1 \).

Since the value \( N \) is a variable, we use the form of average delay and average power consumption. Our goal is to get the optimal collaboration strategy to minimize the value of the function QoE.

\[
\min \{ \frac{1}{N} \sum_{i=1}^{N} [\beta_{QoE} \cdot T_i + (1 - \beta_{QoE}) \cdot E_i] \}
\]

\[s.t. k_{ii} + \sum_{j=1}^{N+1} k_{ij} = 1, \forall i \in \{1, 2, \ldots, N\}\]
ALGORITHM DESCRIPTION

From the mathematical point of view, this problem is a function optimization problem. Our goal is to get the value of each element in the matrix $K$ that minimizes the value of the objective function. Therefore, the number of variables is $N^*(N+1)$. The formula (12) with constraint (13) is a Optimization problem with $N$ equality constraints, it can be solved based on genetic algorithm. We call it cooperation algorithm.

<table>
<thead>
<tr>
<th>TABLE I. COOPERATION ALGORITHM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm 1: cooperation algorithm</td>
</tr>
<tr>
<td>1: Initialization general parameters</td>
</tr>
<tr>
<td>2: Set the equality constraints</td>
</tr>
<tr>
<td>for $i=1:N$</td>
</tr>
<tr>
<td>for $j=1:N+1$</td>
</tr>
<tr>
<td>$k_{ii} + \sum_{j=1}^{N+1} k_{ij} = 1$;</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>3: Set the genetic algorithm parameters</td>
</tr>
<tr>
<td>4: Calling genetic algorithm</td>
</tr>
<tr>
<td>5: Initial definition matrix, Assign values to variables in turn</td>
</tr>
<tr>
<td>for $ii=1:N^*(N+1)$</td>
</tr>
<tr>
<td>$K(ii)=K_{star}(ii)$;</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>6: Get the optimal solution at a specific $N$ value</td>
</tr>
<tr>
<td>7: Calculate average delay and average energy consumption</td>
</tr>
</tbody>
</table>

The general flow of the genetic algorithm is as follows:

<table>
<thead>
<tr>
<th>TABLE II. GENETIC ALGORITHM STEPS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Convert vectors into matrices, initial definitions and assignments</td>
</tr>
<tr>
<td>2: Iteration: Calculation fitness</td>
</tr>
<tr>
<td>Restriction condition acts as a penalty function</td>
</tr>
<tr>
<td>3: Get the best fitness.</td>
</tr>
</tbody>
</table>

The proposed algorithm based on genetic algorithm is described in Algorithm 1.

RESULT EVALUATION

In this section, we introduced the specific setting of simulation parameters, and then, we calculated the average time delay and average energy consumption.
in different scale scenarios, and compared the results with the calculation results of the basic algorithm without the cooperation strategy, and evaluated the simulation results of the collaborative strategy proposed in this paper.

SIMULATION SETUP

Each base station is mounted with a MEC server, so the number of base stations is the number of MEC servers. We initially set the N value to 5, the initial weight to be 0.2, and the ratio of immediate delay/energy consumption to 1:4, Λ obeys Poisson distribution.

The processing power of the MEC server is stronger than the sum of the tasks. In order to reflect the heterogeneity, we use the value of the random number times the task amount as the processing power B of the server, and the generated random number is greater than one. In order to compare the results of different weighting factors with the basic algorithm, the weighting factors $\beta_{QoE}$ are 0.2 and 0.8 respectively to give the fairness of delay and energy consumption.

SIMULATION RESULTS

In order to observe the impact of the value of the number of servers N and the value of the weighting factor on the simulation results in the collaborative strategy, we take $\beta_{QoE}$ value in the previous subsection, the value of N is from 1 to 15, and the step of N is 1.

From the simulation results, it is known that with the increase of N, compared with the basic algorithm, the cooperation algorithm adopting cooperative strategy shows better performance from both delay and energy consumption. This proves that this idea has realized value.

Now let's analyze the impact of the weighting factor. First, we focus on the time delay, when $\beta_{QoE}$ is greater than 0.5, it means the user has more preference for the time delay cost, and the energy consumption requirement is not so high. Therefore, the average delay calculated in this case should be less than the value when $\beta_{QoE}$ is less than 0.5. Observe the simulation results of the delay. When N is small (for this figure, when N is less than or equals to 7), the values under $\beta_{QoE}=0.2$ and $\beta_{QoE}=0.8$ are not much different. When N is gradually increased, most of the bar graphs are The result is small in the case of $\beta_{QoE}=0.8$, which is consistent with the theoretical analysis.

Similarly, when the user has a large emphasis on energy consumption, for this article, $1 - \beta_{QoE}$ should be larger, on the contrary $\beta_{QoE}$ will be smaller, the

![Figure 2. Simulation results.](image)
smaller $R_{QoE}$ should be less energy consumption. The energy consumption results map validates our thinking.

Taking the two results into account, for the same $R_{QoE}$, when the time delay is small, the energy consumption will be larger, which means that the system sacrifices energy for faster processing feedback time, and vice versa.

**CONCLUSION**

In this paper, we proposed a cooperative mobile-edge-computing in Ultra-Dense network system. Then we designed the task workloads, the processing power of the server and the allocation ratio in the cooperation strategy according to the heterogeneity. Next, from the two aspects---time and energy, a computational model is formed. Considering the user experience and user preferences, we add a weighting factor to the objective function. Since this problem is an optimization problem of multi-variable and multi-constraints, we use a cooperative algorithm based on genetic algorithm to perform simulation verification and analyze the results. The results show that the cooperation strategy can be compared with the amount of tasks handled separately between servers. Better reduce latency and energy consumption.

**REFERENCES**

5. Chit Wutyee Zaw, Nway Nway Ei, Han Yeo Reum Im, Yan Kyaw Tun, and Choong Seon Hong, “Cost and Latency Tradeoff in Mobile Edge Computing: A Distributed Game Approach,” IEEE International Conference on Big Data and Smart Computing (BigComp), 2019.
8. Shuming Seng, Xi Li, Hong Ji and Heli Zhang, “Joint Access Selection and Heterogeneous Resources Allocation in UDNs with MEC Based on Non-Orthogonal Multiple Access,” IEEE International Conference on Communications Workshops (ICC Workshops), 2018.