Abstract—In this paper, an offloading algorithm based on Markov Decision Process (MDP) is proposed to solve the multi-objective offloading decision problem in Mobile Edge Computing (MEC) system. The feature of the algorithm is that MDP is used to make offloading decision. The number of tasks in the task queue, the number of accessible edge clouds and Signal-Noise-Ratio (SNR) of the wireless channel are taken into account in the state space of the MDP model. The offloading delay and energy consumption are considered to define the value function of the MDP model, i.e. the objective function. To maximize the value function, Value Iteration Algorithm is used to obtain the optimal offloading policy. According to the policy, tasks of mobile terminals (MTs) are offloaded to the edge cloud or central cloud, or executed locally. The simulation results show that the proposed algorithm can effectively reduce the offloading delay and energy consumption.

Keywords—Markov Decision Process (MDP); offloading; Mobile Edge Computing (MEC); value iteration algorithm

I. INTRODUCTION

With the rapid growth of data demands from intelligent mobile terminals (MTs), there has been an inevitable trend that the computing tasks of mobile terminals are offloaded to the central cloud, which have abundant computing resources [1-2]. Offloading can not only reduce the energy consumption and increase the battery life of mobile terminals, but also reduce the executional delay of tasks. However, since the central cloud is far from users, the offloading delay of the tasks may not satisfy the requirement of users.

To solve the problem, Mobile Edge Computing (MEC) is proposed. The characteristic of MEC is that it pushes edge clouds onto the edge of mobile network, so the edge clouds are more adjacent to users [3]. Although edge clouds own computing resources which are less than the central cloud, they can be deployed in a distributed way and their computing resources can be shared among edge clouds, so they can quickly response to delay-sensitive tasks. In this paper, edge clouds and central cloud are combined to implement offloading.

There have been some studies on offloading in the MEC system in foreign and domestic literatures. Chen et al. [4] studies the multi-user offloading problem in the MEC system and obtains an effective offloading solution with game theory. According to Chen et al. [4], the offloading decision problem among mobile users is considered as a multi-user computational offloading game, and the structural characteristics of the game are analyzed. On this basis, a distributed offloading algorithm is proposed to achieve Nash equilibrium, and the upper bound of the convergence time of the algorithm is deduced. A framework is proposed by Muñoz et al. [5] to complete joint allocation of wireless and computing resources, which can obtain an optimal trade-off between energy consumption and delay. The offloading problem of mobile terminals is regarded as a joint optimization of wireless resources, with the aim to minimize the energy consumption of all terminals and satisfy the delay constraints [6]. In the paper, the optimal solution for one user is deduced and an iterative algorithm is proposed to acquire the optimal solution of the original non-convex problem. A dynamic stochastic offloading algorithm is proposed by Truong-Huu et al. [7] and Markov Decision Process (MDP) is used to obtain the optimal offloading decision policy for mobile users, with the aim to minimize the offloading and executional costs. Huang et al. [8] proposed a dynamic offloading algorithm based on Lyapunov optimization is proposed, which has lower complexity and can
save more energy consumption while satisfying delay constraints. An energy-efficient offloading scheme is proposed by Guo et al. [9] which aims to reduce energy consumption and execution time. In the paper, the offloading problem is transformed into the problem of minimizing energy consumption, while satisfying the constraints of completion time of tasks.

An offloading decision algorithm based on MDP is proposed by Zhang et al. [10]. In this algorithm, the state of the task queue, the status of the task execution and the number of nearby cloudlets are taken into account in the state space. The action space includes two offloading policy: The tasks are executed locally or offloaded to the cloudlets. The cost function of MDP is given and a threshold strategy is used to minimize the cost function. The dynamic resource sharing between mobile users and cloudlets in mobile cloud computing environment is regarded as a semi-Markov Decision Process [11]. Unlike the research results by Zhang et al. [10], the paper chooses the profit of service providers as reward function. The optimization goal is to maximize the profit when satisfying the demand for Quality of Service (QoS), and the problem is solved by transforming the semi-Markov Decision Process into a linear programming problem.

The above offloading algorithms mostly do not consider the combination of edge cloud and central cloud, or the changes in the number of edge clouds accessible to mobile terminals. MDP can be used to obtain the optimal offloading policy because it is a tool for solving optimal decision problems in the random dynamic system based on Markov process theory. It takes random access characteristics of each mobile terminal into consideration, and according to the system state it makes dynamic decisions [10]. Therefore, in this paper, MDP is considered to solve the offloading decision problem in the MEC system.

MDP is the optimal decision process of stochastic dynamic system based on Markov process theory. Compared with some available algorithm mentioned in [7-8], it can consider the random mobility feature of each mobile user and make dynamic decision based on system state, so in this paper, MDP is used to perform offloading decision. Our contributions in this paper are as follows.

1) In our offloading scenario, the multi-layer cloud is considered which consists of the central cloud and edge cloud. The task can be offloaded to edge cloud and the central cloud cooperatively or executed locally.

2) A MDP-based cooperative offloading scheme (MDP-CO) is proposed. The MDP model is developed for offloading decision, whose state space composes the state of the task queue, edge cloud and the central cloud.

The rest of this paper is organized as follows. The system model is introduced in Section II. The state space, action space and the definition of immediate reward function and value function of the MDP model are presented in Section III. The state transition probability is derived in Section IV. Simulation results and analysis are given in Section V and in Section VI, the conclusion is presented.

In this section, the network model for offloading in the MEC system is described firstly. Then the communication model and computational model are given.

A. Network model

The network model for offloading in the MEC system is shown in Fig. 1. Tasks of the mobile terminals can be offloaded to the central cloud, the edge cloud, or directly processed locally. In Fig. 1. There are $N$ edge cloud, one central cloud and $M$ mobile terminals. Each edge cloud is connected to a base station, and all edge clouds are interconnected with wired links and share resources from other edge clouds. Since the edge cloud is closer to the user, the user can directly access the edge cloud via the base station. Therefore, compared to the central cloud, although the edge cloud has limited computing resource, the offloading delay it brings is much shorter. Central cloud is usually located far from users, so users can only first access the base station with wireless links, and then connect to the central cloud via the core network, which usually brings high offloading delay.

![Network model for offloading in the MEC system](image-url)

B. Communication model

In the network model shown in Fig.1, which is a layered model. In the lowest layer is edge cloud, which is mainly deployed with the base stations or access points. In the highest level is the central cloud, which has the largest computing and storage resources, but it is usually far from the users, so the delay brought by offloading applications to the central cloud is largest.

The tasks from mobile terminals can be directly offloaded to the edge cloud with the wireless link. However, if the tasks are offloaded to central cloud, they should firstly be sent to the base station and then offloaded to central cloud via the core network. It is assumed that the $i$-th mobile terminal denoted as $MT_i$ ($i = 1,\cdots,M$), the data size of the input task is denoted as $x_{in}^i$ and the data size of result after processing is denoted as $x_{out}^i$.

If the task is offloaded to the edge cloud, the rate of data transmitted from the mobile terminal to the edge cloud is defined as the uplink rate $r_{u_{i}}$, and the rate of data transmitted
from the edge cloud to the mobile terminal is defined as the downlink rate $r_{ed}$. The round-trip transmission delay of the task is expressed as

$$t_{communicate-e} = \frac{x_{in} + x_{out}}{r_{cu}}$$

(1)

$r_{cu}$ and $r_{ed}$ can be calculated according to the Shannon channel capacity formula, which is given as follows.

$$r_{cu} = \frac{w_{u}^c}{g_{u}^c} \log_2 \left(1 + \frac{g_{u}^c p_{u}^c}{\eta_c}ight)$$

(2)

$$r_{ed} = \frac{w_{d}^c}{g_{d}^c} \log_2 \left(1 + \frac{g_{d}^c p_{d}^c}{\eta_c}ight)$$

(3)

In Eq. (2) and Eq. (3), $w_{u}^c$ and $w_{d}^c$ are uplink and downlink transmission bandwidths of the channel between the terminal and the edge cloud, respectively. $g_{u}^c$ and $g_{d}^c$ are the uplink and downlink transmission gains, respectively. $p_{u}^c$ and $p_{d}^c$ are the uplink and downlink transmission powers and $\eta_c$ is the noise power of the channel. When MT $i$ offloads the task to the edge cloud, the energy consumption of transmission is expressed as follows.

$$\epsilon_{communicate-e} = \frac{x_{in}^{i}}{r_{cu}} p_{u}^{c} + \frac{x_{out}^{i}}{r_{ed}} p_{d}^{c}$$

(4)

If the task is offloaded to the central cloud, the rate of data transmitted from the mobile terminal to the central cloud is defined as the uplink rate $r_{cu}$, the rate of data transmitted from the central cloud to the mobile terminal is defined as the downlink rate $r_{ed}$, and the round-trip transmission delay of the task is expressed as follows.

$$t_{communicate-e} = \frac{x_{in}^{i}}{r_{cu}} + \frac{x_{out}^{i}}{r_{ed}}$$

(5)

In Eq. (5), $r_{cu}$ and $r_{ed}$ can be calculated as follows.

$$r_{cu} = \frac{w_{u}^c}{g_{u}^c} \log_2 \left(1 + \frac{g_{u}^c p_{u}^c}{\eta_c}ight)$$

(6)

$$r_{ed} = \frac{w_{d}^c}{g_{d}^c} \log_2 \left(1 + \frac{g_{d}^c p_{d}^c}{\eta_c}ight)$$

(7)

In Eq. (6) and Eq. (7), $w_{u}^c$ and $w_{d}^c$ are uplink and downlink transmission bandwidths of the channel between the terminal and the central cloud. $g_{u}^c$ and $g_{d}^c$ are the uplink and downlink transmission gains. $p_{u}^c$ and $p_{d}^c$ are the uplink and downlink transmission powers and $\eta_c$ is the noise power. When MT $i$ offloads the task to the central cloud, the energy consumption of transmission is expressed as follows.

$$\epsilon_{communicate-e} = \frac{x_{in}^{i}}{r_{cu}} p_{u}^{c} + \frac{x_{out}^{i}}{r_{ed}} p_{d}^{c}$$

(8)

If the task is executed locally, the delay only includes the processing delay, and the communication delay is equal to zero. The energy consumption also only includes the processing energy consumption, and the transmission energy consumption is equal to zero. Therefore, the transmission delay can be expressed as follows.

$$t_{communicate-e} = \frac{x_{in}^{i}}{r_{cu}} + \frac{x_{out}^{i}}{r_{ed}}$$

(9)

The transmission energy consumption can be expressed as follows.

$$\epsilon_{communicate-e} = \begin{cases} \frac{x_{in}^{i}}{r_{cu}} p_{u}^{c}, & \text{offloaded to edge cloud} \\ \frac{x_{out}^{i}}{r_{ed}} p_{d}^{c}, & \text{offloaded to central cloud} \\ 0, & \text{executed locally} \end{cases}$$

(10)

C. Computational model

Assuming that the speed of the Central Processing Unit (CPU) of the central cloud is $v_c$, the CPU speed of the edge cloud is $v_e$, and the CPU speed of the terminal is $v_t$, the computational delay is calculated as follows.

$$t_{compute} = \begin{cases} \frac{x_{in}^{i}}{v_c} p_{u}^{c}, & \text{offloaded to edge cloud} \\ \frac{x_{out}^{i}}{v_e} p_{d}^{c}, & \text{offloaded to central cloud} \\ \frac{x_{in}^{i}}{v_t} p_{t}^{i}, & \text{executed locally} \end{cases}$$

(11)

Assuming that the CPU power of the central cloud is $p_c$, the CPU power of the edge cloud is $p_e$, and the CPU power of the terminal is $p_t$, the computational energy consumption is calculated as follows.

$$\epsilon_{compute} = \begin{cases} \frac{x_{in}^{i}}{v_c} p_{u}^{c}, & \text{offloaded to edge cloud} \\ \frac{x_{out}^{i}}{v_e} p_{d}^{c}, & \text{offloaded to central cloud} \\ \frac{x_{in}^{i}}{v_t} p_{t}^{i}, & \text{executed locally} \end{cases}$$

(12)

The total offloading delay of the task includes transmission delay and computational delay, which is expressed as follows.

$$t_{total} = t_{communicate} + t_{compute}$$

(13)

The total offloading energy consumption of task includes transmission energy consumption and computational energy consumption, which is expressed as follows.

$$\epsilon_{total} = \epsilon_{communicate} + \epsilon_{compute}$$

(14)

III. STATE TRANSITION PROBABILITY OF THE SYSTEM

In this paper, the system states of the MDP model include the number of tasks in the task queue, the number of edge clouds accessible to mobile terminals and the SNR (Signal-Noise-Ratio) of the wireless channel between central cloud and mobile terminals. The state transition probability of the system will be deduced in this section.

A. Transition probability of the task queue’s state

Assuming that the arrival of new tasks in the task queue of the terminal obeys Poisson process, the arrival rate is $\lambda_d$, and since the queue capacity is limited, it is assumed that the maximum number is $|Q|$. Therefore, during the decision period
The probability of the arrival of $k$ new tasks is

$$f(k, \lambda) = \frac{(\lambda T)^k e^{-\lambda T}}{k!}, k \geq 0$$  \hspace{1cm} (15)$$

The state transition probability of the task queue from $q$ to $q'$ is

$$P(q|q') = \begin{cases} 
1 - \sum_{q''} f(k, \lambda), & q < q' < |q| \\
1, & q = q' = |q| \\
0, & \text{otherwise}
\end{cases}$$  \hspace{1cm} (16)$$

B. Transition probability of the number of accessible edge clouds

Assuming that the number of accessible edge clouds $n$ is regarded as the distribution of Poisson points, $\lambda_n$ represents the spatial density of edge clouds and $L$ represents the distance between the user and the edge cloud. Therefore, the transition probability of the number of accessible edge clouds from $n$ to $n'$ is expressed as follows.

$$P(n|n') = \begin{cases} 
e^{-\lambda_n} (\lambda_n)^n, & n \in \{0, \cdots, N\}, n' \in \{0, N-1\} \\
\sum_{n''} \ne^{-\lambda_n} (\lambda_n)^{n''}, & n' = N
\end{cases}$$  \hspace{1cm} (17)$$

In Eq. (17), $N$ is the maximum number of accessible edge clouds.

C. Transition probability of central cloud's state

In the state of central cloud, the change of SNR in the channel of the wireless link from the user to central cloud is considered. The wireless channel can be regarded as the Rayleigh fading channel [12]. Supposing $s_r$ represents the state of the SNR and the value range of the SNR is $[0, \infty)$, we divide the range into $F$ intervals, that is $[l_0, l_1], [l_1, l_2], \cdots, [l_{F-1}, l_F]$. If $SNR \in [l_l, l_{l+1})$ , the state is $s_r = f , f \in \{0, 1, \cdots, F-1\}$. Therefore, the probability of central cloud’s state from $s_r$ to $s_r'$ is

$$P(s_r|s_r') = \begin{cases} 
0, & |s_r - s_r'| > 1 \\
\frac{N(\Gamma_{s_r, s_r'})}{\pi_{s_r}}, & s_r' = s_r + 1 \\
\frac{N(\Gamma_{s_r, s_r'})}{\pi_{s_r}}, & s_r' = s_r - 1
\end{cases}$$  \hspace{1cm} (18)$$

where,

$$\pi_{s_r} = \exp(-\Gamma_{s_r}) - \exp(-\Gamma_{s_r+1})$$

$$N(\Gamma_{s_r, s_r'}) = \sqrt{2\pi} f_s \exp(-\frac{\Gamma_{s_r}}{\gamma_0})$$  \hspace{1cm} (19)$$

In Eq. (19), $t_p$ represents the time required to transmit a task and $\gamma_0$ represents the average SNR of the wireless channel. In Eq. (20), $f_s$ represents the maximum Doppler frequency shift caused by the mobility of terminals.

D. Transition probability of the system state

Since the transition probability of each state is related to the terminal’s action, the transition probability of the system state is also related to the action. It is assumed that $q, n$ and $s_r$ in the action spaces are independent. $P(q|q', a)$, $P(n|n', a)$ and $P(s_r|s_r', a)$ represent the state transition probability of the task queue, the number of accessible edge clouds, and state of the central cloud, respectively. Therefore, the transition probability of the system state from $s$ to $s'$ can be expressed as follows.

$$P(s'|s, a) = P(q'|q, a)P(n'|n, a)P(s_r'|s_r, a)$$  \hspace{1cm} (21)$$

The values of the parameter in $P(q|q', a)$, $P(n|n', a)$ and $P(s_r|s_r', a)$ are determined by the action of terminals.

IV. MDP MODEL

In this paper, MDP is used to solve offloading decision problem. The MDP model can be represented by a quadruple $M = (S, A, P, \gamma)$. where $S$ is the state space, $A$ is the action space, $P$ is the state transition probability which has been introduced before and $\gamma$ is the reward function. In this section, the definitions of the state space, action space, immediate reward function and value function will be described.

A. State space and action space

In this paper, the state space of the proposed offload algorithm based on MDP is

$$S = \{(q, n, s_r)\}$$  \hspace{1cm} (22)$$

where, $q \in \{0, \cdots, |Q|\}$, $n \in \{0, \cdots, N\}$, $s_r \in \{0, \cdots, F - 1\}$.

The action space of the MDP model is defined as $A = \{0, 1, 2\}$. $a=0$ represents that the task should be executed locally, $a=1$ represents that the task should be offloaded to the edge cloud and $a=2$ represents that the task should be offloaded to the central cloud.

B. Immediate reward function

Defined with the utility function and the cost of offloading delay and energy consumption, the immediate reward function of the MDP model is

$$r(s, a) = \omega_a u(a) - \omega_c c(s, a)$$  \hspace{1cm} (23)$$

where $\omega_u$ and $\omega_c$ are the weight factors for the utility function and cost, respectively. $u(a)$ represents the utility created by user’s action, and is defined as

$$u(a) = \rho \log(1 + a)$$  \hspace{1cm} (24)$$

where $\rho$ is a constant.

The cost is defined as

$$c(s, a) = \omega_d c_{total} + \omega_e e_{total}$$  \hspace{1cm} (25)$$

where $\omega_d$ and $\omega_e$ are the weight factors for the offloading delay and energy consumption, respectively.

C. Value function

The value function is the optimization objective of the MDP model. When the system state is $s$ and the action $a$ is taken, the value function is expressed as follows

$$V(s, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a)V(s', a)$$  \hspace{1cm} (26)$$
In Eq. (26), \( \gamma \) is the discount coefficient. Then, the optimization objective is expressed as

\[
\max_{a \in A} V(s, a), \forall s \in S
\]  

(27)

The solution of Eq. (27) is expressed as

\[
a^* = \arg \max_{a \in A} V(s, a)
\]  

(28)

In Eq. (28), \( a^* \) represents the optimal action that should be taken to maximize the value function.

D. Value iteration algorithm

The solution of Eq. (28) can be acquired with Value Iteration Algorithm. The inputs include the state space \( S \), action space \( A \), the state transition probability, the immediate reward function \( r(s, a) \) and the discount coefficient \( \gamma \). The output is the offloading policy of each system state. The progress of Value Iteration Algorithm is described as follows.

Through the above algorithm, the action \( a^* \) that maximizes the value function can be acquired, so that the optimal offloading policy is obtained. According to the policy, tasks of mobile terminals are offloaded to the edge cloud or central cloud, or executed locally.

Algorithm 1 Value Iteration Algorithm

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>State space ( S ), action space ( A ):</td>
<td></td>
</tr>
<tr>
<td>State transition probability ( P(s</td>
<td>s', a) ):</td>
</tr>
<tr>
<td>Immediate reward function ( r(s, a) ):</td>
<td></td>
</tr>
<tr>
<td>discount coefficient ( \gamma ):</td>
<td></td>
</tr>
</tbody>
</table>

Output:
The optimal offload policy: \( \pi(s) = \arg \max_a \{r(s, a) + \gamma \sum_{s'} P(s|s', a) V^n(s') \} \)

1: Initiate \( V^0 = 0 \), \( n = 0 \);
2: while \( \|V^{n+1} - V^n\| > \frac{\epsilon(1-\gamma)}{2\gamma} \) do
3: \( V^{n+1}(s) = \max_a \{r(s, a) + \gamma \sum_{s'} P(s|s', a) V^n(s') \} \);
4: \( n = n + 1 \);
5: End while

E. PERFORMANCE EVALUATION

In this section, the performance of the proposed offloading algorithm based on MDP is evaluated and compared with other offloading algorithms by computer simulations.

A. Simulation setup

The simulation scenario is shown in Fig. 1. The capacity of the task queue is set to be 2, the maximum number of accessible edge clouds is set to be 2, and the SNR of the wireless channel is quantified into two states, that is \( s_r \in \{0, 1\} \). Other parameters are shown in Table 1.

B. Performance comparison

In this section, the performance of the proposed offloading algorithm based on MDP (MDP) is compared with other algorithms including LOC and RAND. LOC means that the terminal always executes tasks locally. RAND means that the terminal randomly chooses to offload the task to the central cloud or edge clouds, or execute it locally. The performances of algorithms are analyzed through three metrics, including average delay, average energy consumption and average reward.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_u )</td>
<td>200 Mbps</td>
</tr>
<tr>
<td>( w_d )</td>
<td>400 Mbps</td>
</tr>
<tr>
<td>( g_u )</td>
<td>100</td>
</tr>
<tr>
<td>( g_d )</td>
<td>100</td>
</tr>
<tr>
<td>( p_u )</td>
<td>400 mW</td>
</tr>
<tr>
<td>( p_d )</td>
<td>500 mW</td>
</tr>
<tr>
<td>( v_u )</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>( v_d )</td>
<td>200 Mbps</td>
</tr>
<tr>
<td>( p_u )</td>
<td>200 mW</td>
</tr>
<tr>
<td>( p_d )</td>
<td>300 mW</td>
</tr>
<tr>
<td>( v_c )</td>
<td>400 Mbps</td>
</tr>
<tr>
<td>( v_e )</td>
<td>300 Mbps</td>
</tr>
<tr>
<td>( v_t )</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>( p_c )</td>
<td>300 mW</td>
</tr>
<tr>
<td>( p_e )</td>
<td>200 mW</td>
</tr>
<tr>
<td>( p_t )</td>
<td>400 mW</td>
</tr>
</tbody>
</table>

Fig. 2 Average delay
In Fig. 2, average delays of the three algorithms are compared. MDP can effectively reduce the offloading delay. Specifically, MDP reduces the delay by approximately 71% compared with LOC, and approximately 50% compared with RAND.

In Fig. 3, the average energy consumptions of the three algorithms are compared. MDP can effectively reduce the offloading energy consumption. Specifically, MDP reduces energy consumption by approximately 75% compared with LOC and approximately 33% compared with RAND.

In Fig. 4, the average rewards of the three algorithms are compared. The average reward of MDP is greater than LOC and RAND. Specifically, the average reward of MDP is about 3% higher than LOC, and about 52% higher than RAND.

In summary, the offloading algorithm based on MDP has a good performance in terms of average delay, average energy consumption and average reward. It is because the utility function, offloading delay and energy consumption are all considered in the value function of the MDP model. The offloading delay and energy consumption are considered as the reward function of the MDP model and the iteration algorithm always choose the action that minimize the delay and energy consumption. Therefore, the proposed algorithm can always shows best the offloading delay and energy performance.

F. CONCLUSION

In this paper, MDP is applied to solve the multi-objective offloading decision problem in the MEC system. Mobile terminals can offload tasks to the edge cloud or central cloud, or execute them locally. The definition of the state space, action space and value function of the MDP model are given, and the state transition probability is derived. Through the Value Iteration Algorithm, the optimal offloading policy can be acquired that can maximize the value function. The simulation results show that, compared with other algorithms, including LOC and RAND, the proposed offloading algorithm based on MDP can effectively reduce the offloading delay and energy consumption.

References


Bingxin Yao. He was born in 1980. He received the B.S., M.S, and Ph.D. degree in information science and engineering from Southeast University, Nanjing, China in 2002, 2005 and 2009 respectively. From 2010 to now, he was a assistant professor in the Department of Computer Science and Technology, Nanjing technology university, working on various problems in signal processing.

Bin Wu. He was born in Changzhou City, Jiangsu Province in 1974. He graduated from signal and information processing major of Southeast University and obtained a doctorate in engineering in April 2015. From 2005 to now, he has participated in a number of key R & D projects in the communication and power industries and provincial and municipal scientific research projects.

Siyun Wu. He was born in Zhenjiang City, Jiangsu Province in 1994. He graduated from information and communication engineering major of Southeast University and obtained a master's degree in April 2019. Now he works as an operation engineer in the field of cloud computing.

Yin Ji. She was born in Danyang city, Jiangsu Province. She graduated from Southeast University. She received her bachelor's degree from Southeast University in 2008. She began to work in the Internet of things and the Internet in 2016. At present, her main research direction is Internet of things system integration.

Danggui Chen. He was born in 1983 in Xuancheng Country, Anhui Province. He received bachelor's degree from Southeast University in 2005. Now he is an Senior Software Engineer in INRICH TECHNOLOGY. His research interests are Software engineering, automatic control technology, radio monitoring technology.

Limin Liu. She was born in 1995 in Linquan County, Anhui Province. Her major is electronic information engineering. She received a bachelor's degree from Anhui University of Science and Technology in 2018. Now she works as a test engineer in the field of radio frequency.

Contributions of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Bingxin Yao conducted research conception, design and data analysis.

Bin Wu wrote and tested the core algorithm and simulation.

Siyun Wu improved the network model for offloading and gave the analysis of performance of the algorithm.

Yin Ji conducted data collection and manuscript preparation.

Danggui Chen conducted data collation and comparison.

Limin Liu conducted data collection and discussion.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US