

## Research Article

# Data Processing Delay Optimization in Mobile Edge Computing

**Guangshun Li** , **Jiping Wang** , **Junhua Wu** , and **Jianrong Song** 

*School of Information Science and Engineering, Qufu Normal University, Rizhao 276800, China*

Correspondence should be addressed to Guangshun Li; 30752585@qq.com

Received 13 October 2017; Accepted 9 January 2018; Published 6 February 2018

Academic Editor: Shangguang Wang

Copyright © 2018 Guangshun Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of Internet of Things (IoT), the number of mobile terminal devices is increasing rapidly. Because of high transmission delay and limited bandwidth, in this paper, we propose a novel three-layer network architecture model which combines cloud computing and edge computing (abbreviated as CENAM). In edge computing layer, we propose a computational scheme of mutual cooperation between the edge devices and use the Kruskal algorithm to compute the minimum spanning tree of weighted undirected graph consisting of edge nodes, so as to reduce the communication delay between them. Then we divide and assign the tasks based on the constrained optimization problem and solve the computation delay of edge nodes by using the Lagrange multiplier method. In cloud computing layer, we focus on the balanced transmission method to solve the data transmission delay from edge devices to cloud servers and obtain an optimal allocation matrix, which reduces the data communication delay. Finally, according to the characteristics of cloud servers, we solve the computation delay of cloud computing layer. Simulation shows that the CENAM has better performance in data processing delay than traditional cloud computing.

## 1. Introduction

Data storage and computation are two critical problems in cloud computing, which provides a method to solve the limited storage and computing speed of computers or mobile phones [1]. With the development of Internet of Things (IoT), the amount of data transmission shows a trend of exponential increment [2, 3]. It is predicted that the growth trend of data traffic would be eightfold between 2014 and 2020, which will bring a huge challenge for cloud computing [4, 5]. On one hand, limited bandwidth has adverse effects on the efficiency of data transmission. On the other hand, terminal is usually far from the cloud servers and data transmission with long distance increases the transmission delay, which does not meet the requirement of real time, low latency, and high quality of service (QoS) in the network of thousands of IoT devices and affects the overall efficiency of the system [6, 7].

Due to the fact that traditional cloud computing poses many challenges, mobile edge computing (MEC) is proposed, which consists of relatively weak edge devices [8–10]. MEC is a novel paradigm that extends cloud computing capabilities and services to the edge of the network. On one hand, MEC ensures that data processing mainly depends on the local

devices rather than the cloud servers. On the other hand, MEC usually does not need to establish a relationship with remote cloud servers; it can meet most requirements of local users very well [11]. However, MEC does not have enough computing capability compared to cloud servers, since it only includes limited computing devices. Once computing capability of a single edge device is exceeded, MEC requires other edge devices to assist or forward residual data to the cloud servers for processing, so that the system can still maintain good performance with the rapid growth of the number of users or the amount of data.

In this paper, firstly, we propose a novel three-layer network architecture model which combines cloud computing and edge computing (abbreviated as CENAM), and model the communication delay and computation delay of each part of the CENAM. Then, in edge computing, we propose a computational scheme of mutual cooperation between the edge devices based on the weighted undirected graph and use the Kruskal algorithm and Lagrange multiplier method to solve the communication delay and computation delay, respectively. In addition, we focus on the balanced transmission method to solve the data transmission delay from edge devices to cloud servers and solve the computation

delay of cloud servers according to their characters. Finally, based on simulations and numerical results, we show the better performance of CENAM in terms of reducing data processing delay.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we describe the composition of CENAM. Section 4 explains in detail the computational scheme of CENAM that we proposed. Section 5 is algorithm solving. We analyze the performance of our solution in Section 6. In Section 7, we conclude the paper and present some future work.

## 2. Related Work

The related research of MEC has received considerable attention in recent times. For example, Subramanya et al. [12] proposed a resource constrained cloud-enabled small cell that includes a MEC server for deploying mobile edge computing functionalities and presented the architecture with special focus on realizing the proper forwarding of data packets between the mobile data path and the MEC applications. Sharma and Wang [13] put forward a novel framework for coordinated processing between edge and cloud computing by integrating advantages from both the platforms, which can provide real-time information to end users and enhance the performance of wireless IoT networks. Masip-Bruin et al. [14] introduced a layered F2C architecture with benefits and strengths and provided the real need for their coordinated management. To adapt to rapid increment of mobile resources, Lee et al. [15] proposed collaborate platform solution, which greatly simplified the development mechanism of mobile collaborative applications and reduced the delay and energy consumption. Due to the problem that the mobile users are far away from the cloud servers and generate great transmission delay, Intharawijitr et al. [16] relaxed the delay constraint system and selected the edge servers to reduce the system delay effectively. Hu et al. [17] focused on the services allocation problem in MEC and found trade-offs between average network delay and load balance. Simulations showed that the variance of the load on MEC servers is reduced by 18.9% with nearly the same network delay. In order to achieve intelligent D2D communication, Bello and Zeadally [18] focused on intelligent routing protocols and presented an overview of how intelligent D2D communication can be achieved in the IoT ecosystem.

Fog computing [19–21], one of the typical representations of edge computing, has attracted more and more attention to solve related problems. Sarkar and Misra [22] established a theoretical model of fog computing architecture and defined the structural components mathematically. Then, they compared it to the traditional architecture of cloud computing and analyzed the performance of delay and energy consumption in the background of IoT. Deng et al. [23] focused on the problem of delay and optimal energy consumption in cloud-fog computing. They divided the problem of data transmission, existing in fog computing, into three subproblems and analyzed the performance of delay from the perspective of balanced load. But none of them considered the collaboration between edge devices. According to the problem of high

processing delay when cloud computing is used in the medical data scene, He and Ren [24] proposed cloud-fog network architecture model for medical big data. Facing the problem of weak computing capability of edge devices, they put forward a distributed computational scheme. Pham and Huh [25] formulated the task scheduling problem in cloud-fog environment and proposed a heuristic-based algorithm. As a result, they achieved the balance between the makespan and the monetary cost of cloud resources.

## 3. The Description of CENAM

In the real Internet of Things applications, users want to get the information that they want and transmit it as soon as possible, so the information providers need to have high working efficiency. Facing high transmission delay and computing pressure of cloud data center, we propose CENAM, as shown in Figure 1.

Network architecture model is mainly divided into three layers: cloud computing layer, edge computing layer, and mobile terminal layer, where the bottom is mobile terminal layer, which includes all mobile terminal devices, such as smartphones, laptops, and cars. Users accessing the network through intelligent devices not only can quickly access services from the network architecture, but also can store their own information in the data center of network architecture.

The middle layer is edge computing layer, which consists of edge devices (routers, gateways, switches, and access points). Edge devices are mainly distributed in local mobile subscriber premises, such as parks, shopping centers, and buses. Edge computing layer is the bridge between cloud servers and end users. Besides computation and storage of the local data, the residual data that the edge computing layer cannot handle is forwarded to the cloud computing layer for processing.

The top layer is cloud computing layer, which consists of high-end servers and data center. It has powerful computing and storage capabilities and is responsible for computing and storing residual data that the edge computing layer cannot handle.

## 4. Computational Scheme of CENAM

As shown in Figure 1, because the edge devices are close to end users and accept service requests from them through local area network (LAN), the LAN communication delay can be ignored (compared to WAN). Unprocessed data in MEC is forwarded to cloud servers via the WAN for processing, so communication delay from edge devices to cloud servers needs to be considered. In the process of data processing, we mainly consider the delay of edge computing layer and cloud computing layer including communication delay from edge devices to cloud servers.

*4.1. Delay Computational Scheme in Edge Computing Layer.* Edge devices are mainly distributed in areas such as supermarkets, parks, and buses. The cooperation between them is inseparable. In this paper, we abstract a network topology

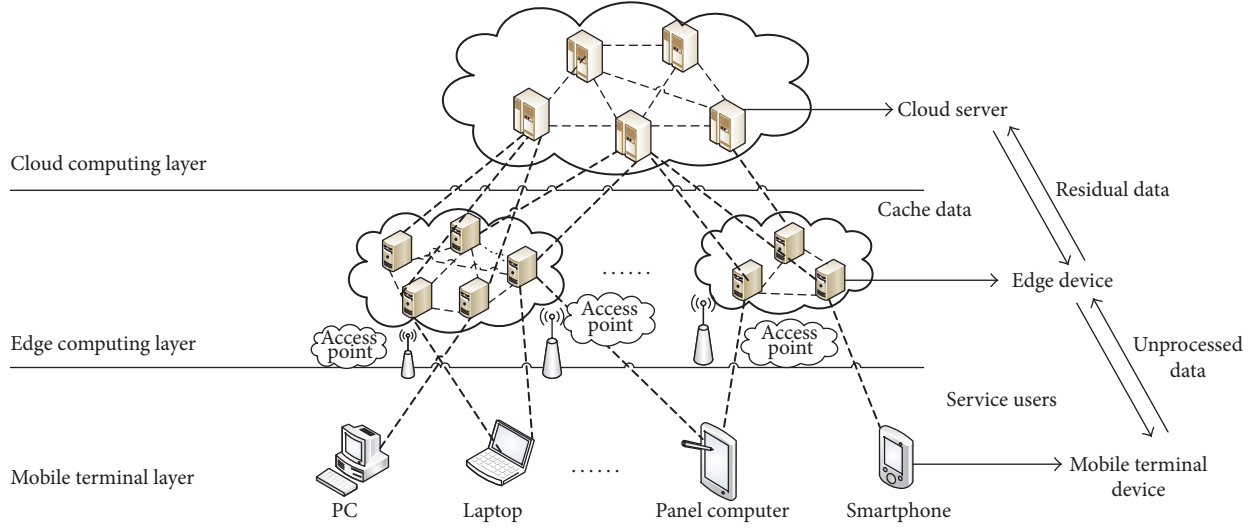


FIGURE 1: CENAM.

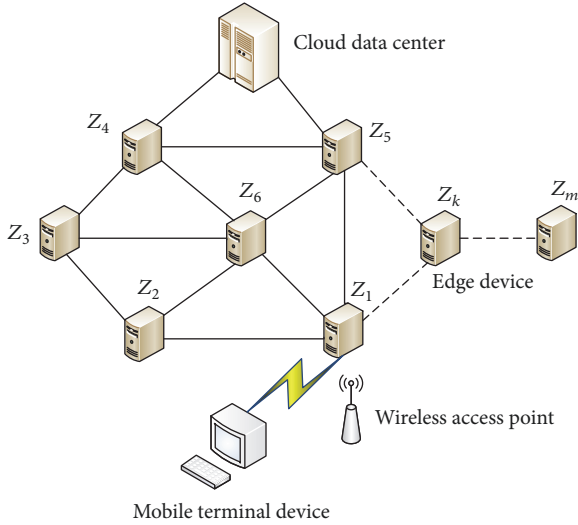


FIGURE 2: Network topology graph of edge devices.

graph consisting of  $m$  edge devices (shown in Figure 2) as a weighted undirected graph (shown in Figure 3).

$$V = \{z_1, z_2, \dots, z_i, \dots, z_m\} \quad (1)$$

is a vertex set, where  $z_i$  is a set of edge devices and  $m$  is the number of edge devices.

$$E = \{e_{z_1, z_2}, \dots, e_{z_i, z_j}, \dots, e_{z_{m-1}, z_m}\} \quad (2)$$

is an edge set, where  $e_{z_i, z_j}$  is the communication link between edge nodes  $z_i$  and  $z_j$ .  $\tau_{z_i, z_j}$  on the edge is the communication delay between edge nodes  $z_i$  and  $z_j$ , and the size of its value is randomly generated with reference to the numerical value of [24].

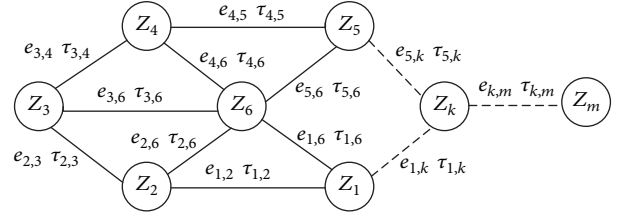


FIGURE 3: Weighted undirected graph of edge devices.

We suppose that the computing capability of each edge node  $z_i$  in Figure 3 is

$$v_{z_i} \quad (i = 1, 2, \dots, m) \quad (3)$$

and the computation task of edge computing layer is  $X$ . In order to reduce the amount of data forwarded from edge computing layer to cloud computing layer to reduce communication delay, we need to enhance the computing capability of edge node and process data on edge network as much as possible. Therefore, we propose a scheme of cooperation between edge devices. In the process of data processing, edge device  $z_i$  receives the computation task  $X$  from end users and divides the task into several subtasks that meet  $x_i = \delta_i X$  and then assigns the subtasks to edge nodes which included themselves to process; the specific scheme is as follows.

The delay of edge computing layer includes the communication delay between edge nodes and the computation delay of edge nodes. For the communication delay, in weighted undirected graph consisting of edge nodes, we regard the communication delay between edge nodes as the weight and use the Kruskal algorithm to generate a minimal spanning tree, so we get the minimum weight  $W(T)$ , that is, the minimum communication delay. For edge device  $i$ , its computation delay can be described by the amount of computation  $x_i$  assigned to it, which satisfies two points: (i)

As the amount of computation increases, the computation delay of edge devices increases accordingly. (ii) The more the amount of computation increases, the faster the computation delay of edge devices increases. Therefore, the following function is used to describe the computation delay of edge device  $i$ .

$$D_i^{\text{Edge}} = \frac{1}{v_{z_i}} a_i x_i^2, \quad (4)$$

where  $v_{z_i}$  is the computing capability of edge device  $i$ ,  $x_i$  is the amount of data to be processed by edge device  $i$ , and  $a_i$  is a predetermined real number between 0 and 1.

So, the computation delay of  $m$  edge nodes can be expressed as

$$D_{\text{Edge}}^{\text{com}} = \min \sum_{i=1}^m \frac{1}{V_{Z_i}} a_i x_i^2 \quad (5)$$

$$\text{s.t.} \quad \sum_{i=1}^m x_i = X \quad 0 < x_i < x_i^{\max},$$

where  $x_i^{\max}$  is the maximum amount of data that edge devices  $i$  can handle and  $X$  is the total amount of data that the edge computing layer needs to process. So the amount of data processed on edge computing nodes can form an  $m$ -dimension vector

$$X = [x_1, x_2, \dots, x_m]. \quad (6)$$

Thus, the following objective function is established for the delay generated in edge computing layer:

$$D_{\text{Edge}} = \min \left( \sum_{i=1}^m \frac{1}{V_{Z_i}} a_i x_i^2 \right) + W(T) \quad (7)$$

$$\text{s.t.} \quad \sum_{i=1}^m x_i = X \quad 0 < x_i < x_i^{\max}.$$

**4.2. Delay Computational Scheme in Cloud Computing Layer.** Delay in cloud computing layer includes the computation delay of cloud servers and the communication delay from edge devices to cloud servers. In cloud computing layer, we assume that there are  $n$  cloud servers; the amount of data processed on cloud servers is  $Y$ . For the cloud server  $j$ , if the amount of data it needs to deal with is  $y_j$  and the computing capability is  $v_j$ , its computation delay can be expressed as

$$D_j^{\text{com}} = \frac{y_j}{v_j} \quad (j = 1, 2, \dots, n), \quad (8)$$

So, when the  $n$  cloud servers deal with the amount of data being  $Y$ , the computation delay in cloud computing layer is

$$D_{\text{cloud}}^{\text{com}} = \sum_{j=1}^n \frac{y_j}{v_j} \quad (j = 1, 2, \dots, n) \quad (9)$$

$$\text{s.t.} \quad \sum_{j=1}^n y_j = Y \quad y_j \geq 0,$$

If the data loss rate is not considered, the delay of the WAN transmission path from the edge device  $i$  to the cloud server  $j$  is  $d_{ij}$ , and the traffic rate is  $\lambda_{ij}$ . According to [23], the communication delay is

$$D_{ij}^{\text{comm}} = d_{ij} \lambda_{ij} \quad (i = 1, 2, \dots, m \quad j = 1, 2, \dots, n). \quad (10)$$

Therefore, the delay of transmitting data from edge devices to cloud servers in CENAM can be expressed as

$$D_{\text{Edge-Cloud}}^{\text{comm}} = \sum_{i=1}^m \sum_{j=1}^n d_{ij} \lambda_{ij} \quad (i = 1, 2, \dots, m \quad j = 1, 2, \dots, n)$$

$$\text{s.t.} \quad \sum_{i \in M} \lambda_{ij} = y_j \quad (11)$$

$$\sum_{j \in n} \lambda_{ij} = x_i$$

$$0 \leq \lambda_{ij} \leq \lambda_{ij}^{\max},$$

where  $\lambda_{ij}^{\max}$  is the maximum traffic rate limited by bandwidth.

**4.3. Problem Conception.** In the CENAM, system delay mainly includes the computation delay of edge devices and cloud servers and the communication delay from edge devices to cloud servers, so the system delay is defined as

$$D^{\text{sys}} = D_{\text{Edge}} + D_{\text{Cloud}}^{\text{com}} + D_{\text{Edge-Cloud}}^{\text{comm}}. \quad (12)$$

Considering the delay optimization of CENAM and the system load balancing, the objective function is established as

$$\min D^{\text{sys}} \quad (13)$$

$$\text{s.t.} \quad L = X + Y$$

$$\sum_{i=1}^m x_i = X$$

$$\sum_{j=1}^n y_j = Y,$$

where  $L$  is the total amount of data processed in system,  $X$  is the amount of data processed in edge computing layer, and  $Y$  is the amount of data processed in cloud computing layer, and they meet  $L = X + Y$ .

## 5. Algorithm Solution

### 5.1. Solution of Delay Optimization in Edge Computing Layer

**5.1.1. Solution of Communication Delay in Edge Computing Layer.** We use Kruskal algorithm to solve the problem of minimum communication delay between edge node. For the weighted undirected graph  $G = (V, E)$ , its minimum spanning tree is

$$T = (U, TE), \quad (14)$$



and the initial state is

$$\begin{aligned} U &= V, \\ TE &= \{\}. \end{aligned} \quad (15)$$

In this way, each edge node in  $T$  constitutes a connected component. Then, according to the order of the weight of edges from small to big, the edges of set  $E$  are investigated in turn. If the two vertices under investigation are two different connected components in  $T$ , the investigated edges are added to  $TE$ , and the two connected components are joined to one. If not, the edge is removed to avoid the loop. So moving on, when the connected component in  $T$  is one, it is the minimal spanning tree of  $G$ . Kruskal algorithm is described as follows:

- (1) Initialization:  $U = V$ ;  $TE = \{\}$ .
- (2) Repeat the following operations until the number of connected components in  $T$  is one.  
Find the shortest edge  $(u, v)$  in  $E$ .  
If vertices  $u$  and  $v$  are located in two different connected components in  $T$ , then
  - (i) incorporate edge  $(u, v)$  into  $TE$ ,
  - (ii) merge two connected components into one.
 In  $T$ , we mark edge  $(u, v)$  and make it not participate in the selection of the subsequent minimal edges.
- (3) Finally, we get the minimum communication delay  $W(T)$ .

**5.1.2. Solution of Computation Delay in Edge Computing Layer.**  
The target function of computation delay is

$$\begin{aligned} f(x) &= \min \sum_{i=1}^m \frac{1}{V_{z_i}} a_i x_i^2, \\ h_i(x) &= \begin{cases} 0 < x_i < x_i^{\max} \\ \sum_{i=1}^m x_i = X. \end{cases} \end{aligned} \quad (16)$$

We solve this equality constrained optimization problem based on the Lagrange multiplier method, and the specific steps are as follows.

(1) Given the initial point  $x^{(0)}$ , the initial multiplier vector  $v$  ( $v_1, v_2, \dots, v_m$ ), the penalty factor  $M > 0$ , the amplification factor  $a > 0$ , and the accuracy  $\varepsilon > 0$ , the parameter  $\gamma \in (0, 1)$  and  $k = 1$ .

(2) Construct an objective function as follows:

$$F(x) = f(x) + \frac{M}{2} \sum_{i=1}^m h_i(x)^2 - \sum_{i=1}^m v_i h_i(x). \quad (17)$$

$f(x)$  is target function and  $h_i(x)$  is constraint function.

(3) Use the unconstrained nonlinear programming method (Newton method is used in this paper),  $x_{(k-1)}$  is used as the initial point to solve  $\min F(x)$ , and the optimal solution is  $x_{(k)}$ .

(4) If

$$\|h(x_{(k)})\| < \varepsilon, \quad (18)$$

then stop iterating and output  $x_{(k)}$ ; otherwise go to step (5).

(5) If

$$\frac{\|h(x_{(k)})\|}{\|h(x_{(k-1)})\|} \geq \gamma, \quad (19)$$

then  $M = \alpha M$ ; otherwise  $M$  remains unchanged; go to step (6).

(6) Define

$$v_i = v_i - M h_i(x_{(k)}), \quad i = 1, 2, \dots, m, \quad (20)$$

set  $k = k + 1$ , and go to step (2).

So, we obtain the optimal solution  $x_{(k)}$  and the minimum computation delay of edge computing layer. And the optimal delay of the edge computing layer is the sum of the computation delay and communication delay.

## 5.2. Solution of Delay Optimization in Cloud Computing Layer.

According to the above analysis, the delay of cloud computing layer mainly includes the computation delay of cloud servers and the communication delay from edge devices to cloud servers. Based on the load balancing principle, the communication delay problem is regarded as a balanced transmission problem. We suppose that the amount of data that is not processed in edge computing layer (i.e., the amount of data that needs to be processed in cloud computing layer) is  $l_i$  and the amount of data that each cloud server can deal with is  $y_j$ . The amount of data that is not processed in edge computing layer needs to be forwarded to cloud computing layer for processing. In order to minimize the transmission delay, an optimal transmission scheme is needed. The delay of the WAN transmission path from the edge device  $i$  to the cloud server  $j$  is  $d_{ij}$ , and the following delay matrix can be obtained:

$$d_{ij} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \dots & \dots & & \dots \\ d_{m1} & d_{m2} & & d_{mn} \end{bmatrix}. \quad (21)$$

$\lambda_{ij}$  is the traffic rate from edge device  $i$  to cloud server  $j$ , and the following transmission matrix can be obtained:

$$\lambda_{ij} = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2n} \\ \dots & \dots & & \dots \\ \lambda_{m1} & \lambda_{m2} & & \lambda_{mn} \end{bmatrix}. \quad (22)$$

Differing from [23], we use the method of balanced transmission to solve the problem of communication delay, and obtain the optimal allocation matrix  $Z_{ij}$ .  $Z_{ij}$  is the

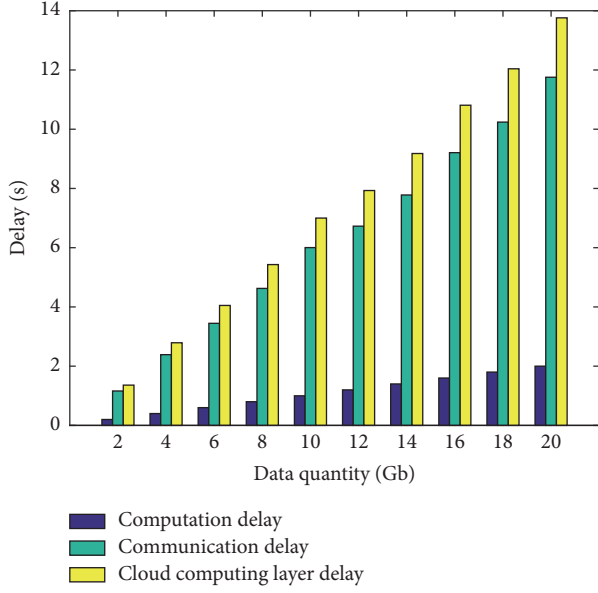


FIGURE 4: Computation and communication delay of cloud computing layer.

amount of transmission data from edge device  $i$  to cloud server  $j$ .

$$Z_{ij} = \begin{bmatrix} Z_{11} & Z_{12} & \cdots & Z_{1n} \\ Z_{21} & Z_{22} & \cdots & Z_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ Z_{m1} & Z_{m2} & \cdots & Z_{mn} \end{bmatrix}. \quad (23)$$

For the computation delay of cloud servers, we do not consider the data loss rate. The principle of conservation of  $\sum_{i \in M} \lambda_{ij} = y_j$  shows that the amount of data that needs to be processed by cloud server is  $y_j$ . In experiment, we assume that the number of cloud servers  $n$  is 5 and the computing capability of cloud server is 10 GHz. We adopt MATLAB experimental platform, and the data in experiment is all simulated. We obtain the communication delay and computation delay by MATLAB simulation, as shown in Figure 4.

Figure 4 shows that, in cloud computing layer, the computation delay is relatively small and varies little because of the strong computing capability of cloud server. The delay of cloud computing layer is mainly caused by the communication delay from edge devices to cloud servers. On one hand, this is because the communication with long distance from end users to the cloud data center may generate high delay. On the other hand, the limitation of network bandwidth increases the transmission delay from edge devices to cloud servers greatly. As the amount of data increases continuously, the communication delay increases faster.

## 6. Simulation Results and Analysis

In order to verify the effectiveness of the scheme of data processing delay in edge computing layer, we compare the

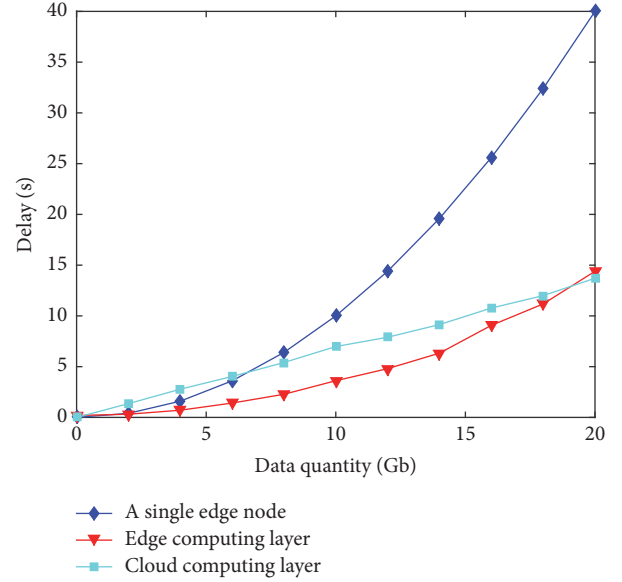


FIGURE 5: Comparison of data processing delays.

data processing delay performance to a single edge node and the cloud computing layer at first. Then, we analyzed the impact of the proportion of the amount of data  $a$  processed in edge computing layer on data processing delay. Finally, the influence of the number of edge nodes on data processing delay is analyzed.

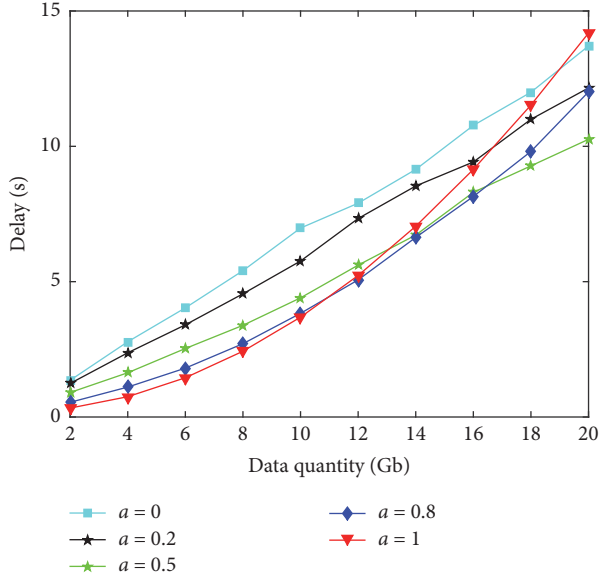
The experimental platform adopts MATLAB, and the computing capability and communication delay of edge devices and cloud servers are all found in [24]. The data in experiment is set by simulation, the number of edge nodes  $m$  is 10, and the number of cloud servers  $n$  is 5. Besides, we set the computing capability of cloud servers to 10 GHz. The computing capability of each edge device is shown in Table 1.

**6.1. Performance Analysis of Data Processing Delays in Edge Computing Layer.** In edge computing layer, we propose the scheme of computation delay and communication delay. In order to verify its effectiveness in data processing delay, we compare the delay to a single edge node and cloud computing layer. The experimental results are shown in Figure 5.

Experimental results show that when the amount of data is  $X < 2$  Gb, a single edge node has less delay compared to the cloud computing layer and the edge computing layer; this is because it does not produce communication delay, and the amount of data is within the range of computing capability of a single edge node. However, as the amount of data increases, the delay caused by the computing capability of the single edge node will increase rapidly. Cloud servers have strong computing capability, but the end users are far away from them and limited by bandwidth, which will generate great communication delay, so the delay of data processing is higher than that of edge computing layer. At this time, the scheme of cooperation between multiple edge nodes in edge computing layer shows better performance. When the amount of data is  $X > 19$  Gb, the delay in edge computing layer will be affected by the limitation of computing capability of

TABLE 1: The computing capability of edge nodes.

Parameter	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$	$z_7$	$z_8$	$z_9$	$z_{10}$
$v_i/\text{GHz}$	2	1	1	0.5	0.4	0.7	0.6	0.3	0.5	1

FIGURE 6: The impact of  $a$  on data processing delay.

the single edge node, and there is a significant rise in data processing delay. However, cloud computing layer with its powerful computing capability makes data processing delay less than that of edge computing layer. Therefore, it is possible to put the appropriate amount of data to edge computing layer to reduce the delay.

**6.2. The Impact of  $a$  on Data Processing Delay.**  $a$  is the percentage of data that is processed in edge computing layer. To verify the performance of CENAM, we study the impact of  $a$  on data processing delay. Simulation is shown in Figure 6.

The experimental results show that when  $a \leq 0.5$ , that is, the amount of data processed in edge computing layer is less than half, the larger the  $a$ , the smaller the delay. When  $a > 0.5$  and the amount of data is small, the larger the  $a$ , the smaller the delay. However, as the amount of data increases, the delay will increase accordingly. And the larger the  $a$ , the faster the corresponding delay, and it even exceeds the traditional cloud computing layer delay. This is because when the amount of data reaches a threshold (e.g., when  $a = 0.8$ , the threshold is 16 Gb; when  $a = 1$ , the threshold is 13 Gb), data processing delay increases rapidly and exceeds a certain value because of the limitation of computing capability of the single edge node. This shows that data processing delay in edge computing layer is limited by computing capability of the single edge node, and when the amount of data increases to a certain degree, the delay will increase. It also shows that when the total amount of data is small, we can process them in edge computing layer and generate small delay. However, when there is a large

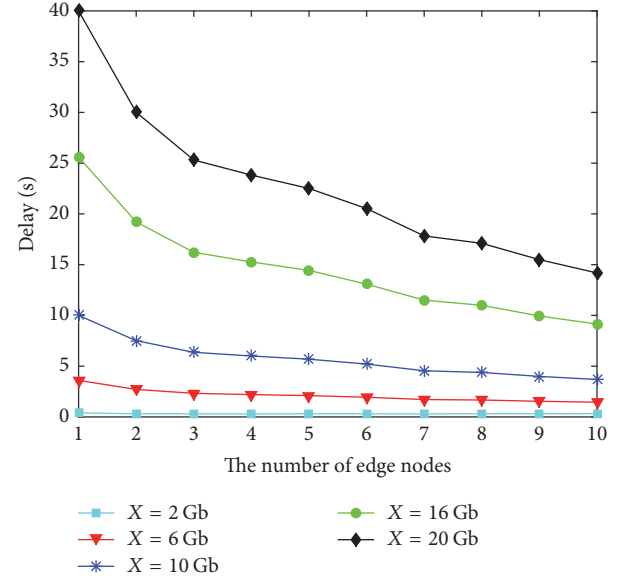


FIGURE 7: The influence of the number of edge nodes on data processing delay.

amount of data, the cooperation between edge devices and cloud servers can effectively reduce the data processing delay.

**6.3. The Influence of the Number of Edge Nodes on Data Processing Delay.** In order to study the influence of the number of edge nodes on data processing delay in edge computing layer, we solve the data processing delay when the total amount of data is 2 Gb, 6 Gb, 10 Gb, 16 Gb, and 20 Gb. The result is shown in Figure 7.

The experimental results show that, with the increase of the number of edge nodes, the overall data processing delay shows a downward trend. When the amount of data is small (as shown below 6 Gb), the increase in the number of edge nodes has less impact on data processing delay, and it is basically stationary. When the amount of data is large (such as from 10 Gb to 20 Gb), with the increase of the number of edge nodes, there is a significant decrease in data processing delay. That is because when the amount of data is small, the computation delay of edge nodes is small, and the communication delay will become large with the increase of edge nodes. So the data processing delay mainly depends on the communication delay between edge nodes, but the change is not obvious. When the amount of data is large, the computation delay of edge nodes increases accordingly, and the communication delay between edge nodes is relatively stable when the bandwidth is allowed. The delay of data processing mainly depends on the computation delay of edge nodes. Therefore, as the number of edge nodes increases, the amount of data processed by each edge node will decrease

and the delay will also be reduced, so that the overall data processing delay will decrease significantly. This shows that, according to the computing capability of edge nodes, it is important to determine the appropriate number of edge nodes to reduce the delay of data processing.

## 7. Conclusion and Future Work

In this paper, the concept of CENAM is proposed to solve the problem of high delay of data processing in traditional cloud computing. In edge computing layer, we use the method of the cooperation between multiple edge nodes to improve data processing capability and reduce the computation delay of edge computing layer. Besides, we use the Kruskal algorithm to solve the communication delay between edge nodes. In cloud computing, the communication delay from edge devices to cloud servers is reduced based on the way of balanced transmission. Simulation shows that the CENAM proposed in this paper can effectively reduce the data processing delay and perform better than both the single edge node and traditional cloud computing. In the future work, we will continue to study the influence of the location allocation and service mode of edge devices on data processing delay and energy consumption, so that we can further improve the performance of system.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (61672321 and 61771289), the Shandong Provincial Graduate Education Innovation Program (SDYY14052 and SDYY15049), the Shandong Provincial Specialized Degree Postgraduate Teaching Case Library Construction Program, the Shandong Provincial Postgraduate Education Quality Curriculum Construction Program, the Shandong Provincial University Science and Technology Plan Projects (J16LN15), and the Qufu Normal University Science and Technology Plan Projects (xkj201525).

## References

- [1] W. Fang, "Paradigm shift from cloud computing to fog computing," *Journal of Nanjing University of Information Science Technology*, vol. 8, no. 5, pp. 404–414, 2016.
- [2] A. J. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.
- [3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [4] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: Progress and challenges," in *Proceedings of the 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2016*, pp. 83–84, UK, April 2016.
- [5] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "Threats to Networking Cloud and Edge Datacenters in the Internet of Things," *IEEE Cloud Computing*, vol. 3, no. 3, pp. 64–71, 2016.
- [6] S. C. Li, "Research on Internet of Things Architecture Based on Fog Computing," *Telecommunication Engineering Technology and Standardization*, vol. 30, no. 11, pp. 63–66, 2017.
- [7] P. Corcoran and S. K. Datta, "Mobile-Edge Computing and the Internet of Things for Consumers: Extending cloud computing and services to the edge of the network," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 73–74, 2016.
- [8] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2017.
- [9] M. Satyanarayanan, "The emergence of edge computing," *The Computer Journal*, vol. 50, no. 1, Article ID 7807196, pp. 30–39, 2017.
- [10] J. Tang and T. Q. S. Quek, "The role of cloud computing in content-centric mobile networking," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 52–59, 2016.
- [11] S. K. Datta and C. Bonnet, "An edge computing architecture integrating virtual IoT devices," in *Proceedings of the 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, pp. 1–3, NAGOYA, Japan, October 2017.
- [12] T. Subramanya, L. Goratti, S. N. Khan, E. Kafetzakis, I. Gianoulakis, and R. Riggio, "A practical architecture for mobile edge computing," in *Proceedings of the 2017 IEEE Conference on Network Function Virtualization and Software-Defined Networks (NFV-SDN)*, pp. 1–4, Berlin, November 2017.
- [13] S. K. Sharma and X. Wang, "Live Data Analytics with Collaborative Edge and Cloud Processing in Wireless IoT Networks," *IEEE Access*, vol. 5, pp. 4621–4635, 2017.
- [14] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Communications Magazine*, vol. 23, no. 5, pp. 120–128, 2016.
- [15] J. Lee, H. Lee, Y. C. Lee, H. Han, and S. Kang, "Platform Support for Mobile Edge Computing," in *Proceedings of the 2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, pp. 624–631, Honolulu, CA, USA, June 2017.
- [16] K. Intharawijitr, K. Iida, H. Koga, and K. Yamaoka, "Practical Enhancement and Evaluation of a Low-Latency Network Model Using Mobile Edge Computing," in *Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, pp. 567–574, Turin, July 2017.
- [17] B. Hu, J. Chen, and F. Li, "A dynamic service allocation algorithm in mobile edge computing," in *Proceedings of the 2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 104–109, Jeju Island, Korea (South), October 2017.
- [18] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal*, 2014.
- [19] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, "Fog Computing: Focusing on Mobile Users at the Edge," *Computer Science*, pp. 1–7, 2015.
- [20] I. Stojmenovic and S. Wen, "The fog computing paradigm: scenarios and security issues," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '14)*, pp. 1–8, IEEE, Warsaw, Poland, September 2014.



- [21] Z. H. Yang, "Boundary computation model of Internet of things: fog computing," *Internet of things technology*, vol. No. 12, pp. 65–67, 2014.
- [22] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications," *IET Networks*, vol. 5, no. 2, pp. 23–29, 2016.
- [23] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, 2016.
- [24] L. X. He and Z. Y. Ren, "Cloud network for medical large data and its distributed computing scheme," *Journal of Xi'an Jiaotong University*, vol. 50, no. 10, pp. 71–77, 2016.
- [25] X.-Q. Pham and E.-N. Huh, "Towards task scheduling in a cloud-fog computing system," in *Proceedings of the 18th Asia-Pacific Network Operations and Management Symposium, APNOMS 2016*, Japan, October 2016.