

Research Article

Energy-Efficient Cloudlet Management for Privacy Preservation in Wireless Metropolitan Area Networks

Xiaolong Xu ^{1,2,3} Rui Huang,^{1,2} Ruihan Dou,⁴ Yuancheng Li,^{1,2} Jie Zhang ³,
Tao Huang,⁵ and Wenbin Yu^{1,2}

¹School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China

²Jiangsu Engineering Centre of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, China

³State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

⁴Jinling High School, Nanjing, China

⁵Silicon Lake College, Kunshan, China

Correspondence should be addressed to Jie Zhang; njujiezhang@gmail.com

Received 30 April 2018; Accepted 1 August 2018; Published 24 September 2018

Academic Editor: David Megias

Copyright © 2018 Xiaolong Xu 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.

Currently, cloudlet based wireless metropolitan area network (WMAN) is emerging as an effective paradigm to extend the performance of mobile devices, which enables the execution of computational intensive mobile applications. But the normal operation of cloudlets consumes a large amount of energy, which brings about carbon dioxide emissions, aggravation of the greenhouse effect, etc. Meanwhile the data transmission of various mobile users among cloudlets would cause leakage of personal privacy. In view of this challenge, we propose an energy-efficient cloudlet management method, named ECM, for privacy preservation in WMAN. Technically, an optimization model is designed to formalize our problem. Then, based on live virtual machine (VM) migration technique, a corresponding privacy-aware VM scheduling method for energy saving is designed to determine which VMs should be migrated and where they should be migrated. Finally, experimental data demonstrate that the proposed method is both efficient and effective.

1. Introduction

In recent years, wireless metropolitan area network (WMAN) is emerging as a public network, which covers the metropolis and provides abundant services for human beings in urban cities [1]. Mobile devices in WMAN ubiquitously access rich resources from remote data centers via access points (APs). With the popularity and development of mobile devices, the demands of mobile users for wireless broadband are constantly increasing. WMAN is useful and powerful to satisfy such increasing resource requirements [2, 3].

In practice, as the service requirements of mobile users increase, mobile applications, such as interactive gaming, virtual reality, and natural language processing, are becoming more computation-intensive. But the computing capacity of mobile devices remains limited, due to the service experience of the mobile devices about weight, size, and battery life [4].

A powerful and effective way to extend the performance of mobile devices is to offload some loads on the mobile devices to the remote clouds for execution. Most existing researches in computation offloading tend to regard the cloud as the remote offloading destination, due to its abundant computing resources [5]. However, the cloud generally is located remotely and far away from the mobile users, which leads to the unneglectable and unavoidable communication delay between the mobile device and the remote cloud. To reduce such communication delay, cloudlets, which are some small mobile data centers deployed in WMAN, are introduced in the service framework of WMAN [6, 7].

There are a large amount of APs collocated with cloudlets in WMAN, where mobile devices are available to access the cloudlet resources easily, provided in the form of virtual machines (VMs) [8]. The obvious advantage of the cloudlet, compared to cloud, is the close physical distance between

the mobile devices and the cloudlet, which shortens the communication delay and improves the experience of mobile users. In such a cloudlet based WMAN, users can deploy computational intensive mobile applications to cloudlets nearby, such as resolving HD video, real-time large-scale online games, and real-time navigation. However, when the cloudlets deployed in WMAN are placed incorrectly or many mobile users concurrently access the cloudlet resources in a region, high communication delay which affects the experience of mobile users would be generated in WMAN [9, 10]. Therefore, it is necessary to pay attention to the communication delay for cloudlet management.

On the other hand, in order to support the efficient and effective operation of each cloudlet deployed in WMAN, a certain amount of energy is necessary to be consumed by the computing nodes and the cooling systems in the cloudlets [11, 12]. However, when the user traffic of cloudlets in some areas of urban cities is low, the usage of cloudlet resources would decrease. And if these cloudlets keep operating, excessive energy consumption continues to grow, which brings about some effects in urban cities, such as increased carbon dioxide emissions and aggravating the greenhouse effect [13, 14]. As a result, it is of great importance to implement cloudlet management for WMAN in an energy-efficient way.

Live VM migration technique provides potential and opportunities for cloudlet management to reduce energy consumption. Live VM migration is of great benefit to improve the usage of cloudlet resources, which does a great favor to reduce the number of running cloudlets and in turn achieve the energy efficiency. Consider the scenario that a cloudlet near a scenic spot is heavily loaded during the daytime, but this cloudlet has little load at night, so that these running VMs on this cloudlet could be migrated to another cloudlet and turn this cloudlet off to reduce energy consumption.

Moreover, due to the continuous evolution of WMAN and increasing diverse set of mobile users, WMAN has significant privacy problems [1]. When using the cloudlets to extend the performance of mobile devices, multiple mobile services with the personal private information could be accessed by multiple cloudlets due to the VM migrations. This information is very sensitive, as users' preferences, habits, and interests could be inferred from them. Especially in an open network such as WMAN, personal information is easily leaked with high probability. The more nodes data pass through in WMAN when the data about users' privacy is transmitted, the more possibly users' privacy could be leaked [15]. To realize privacy preservation of mobile users, it is necessary to reduce the data transmission hops due to VM migrations [16].

With these observations, it is still a challenge to realize energy-efficient resource management for cloudlets, while considering the privacy preservation due to VM migrations. In view of this challenge, a VM scheduling method for cloudlets in WMAN, to save energy and protect privacy, is proposed in this paper [17]. Our main contributions are threefold. Firstly, an optimization model for energy consumption of VM migrations in the cloudlets is designed. Secondly, taking into account the transmission delay and

the possibility of privacy leakage, an energy-efficient cloudlet management method, named as ECM, is proposed to achieve the goal of energy saving and privacy preservation, through VM migrations. Specifically, ECM consists of five main steps, i.e., the destination cloudlet identification, shortest time identification, VM migration scheduling strategy designing, idle-state detection for cloudlets, and global VM scheduling. Finally, comprehensive experiments and simulations are conducted to demonstrate the validity of our proposed method.

The structure of this paper is organized as follows. Section 2 shows the analysis of the energy consumption of cloudlets and privacy preservation in WMAN. An effective method of VM scheduling to save energy and privacy preservation is illustrated in Section 3. Section 4 presents the experiment results. Section 5 reviews the related work. Section 6 concludes the paper and gives an outlook of the future work [18].

2. Preliminary Knowledge

In this section, we introduce some formalized concepts and definitions of the cloudlet management in WMAN. Besides, we analyze the potential energy consumption and privacy preservation in the paper. Key terms and descriptions used in this paper are listed in Table 1.

2.1. Basic Concepts and Definitions. In WMAN, mobile users access the computation and storage resources through the access points (APs) and offload their computing tasks to the cloudlets for resource response [19].

Suppose that there are N cloudlets in the WMAN, denoted as $C = \{c_1, c_2, \dots, c_N\}$ and there are M mobile tasks running on the cloudlets, denoted as $K = \{k_1, k_2, \dots, k_M\}$. Let W be the amount of the APs deployed in the WMAN, and the AP set is denoted as $A = \{a_1, a_2, \dots, a_W\}$. Figure 1 shows an example of WMAN framework with five APs (i.e., $a_1, a_2, a_3, a_4,$ and a_5) and three cloudlets (i.e., $c_1, c_2,$ and c_3). In this example, c_1 is connected to a_1 , c_2 is connected to a_2 , and c_3 is connected to a_3 . Mobile users access resources from the cloudlets c_1 and c_2 via the APs a_1 and a_2 . Without loss of generality, we assume that there is only one physical machine (PM) deployed in a cloudlet. Let g_n be the number of VM instances accommodated in the n th ($n = \{1, 2, \dots, N\}$) cloudlet c_n .

Some basic concepts for further energy-aware VM scheduling method design about cloudlet management in WMAN are presented as follows.

Definition 1 (VM instance requirement of t_m). The VM instance requirement of t_m is defined by the requested number of VM instances, denoted as r_m .

Definition 2 (composed VM for t_m). The VM instances in the same cloud for executing the same task t_m could be composed as a special VM, denoted as s_m .

As the tasks are deployed on the VMs among the cloudlets, there are M running composed VMs, denoted as $S = \{s_1, s_2, \dots, s_M\}$.

TABLE I: Key terms and descriptions.

Terms	Description
C	The set of running cloudlets, $C = \{c_1, c_2, \dots, c_N\}$
K	The set of tasks running on the cloudlets, $K = \{k_1, k_2, \dots, k_M\}$
A	The set of APs deployed in WMAN, $A = \{a_1, a_2, \dots, a_W\}$
$E(t)$	The total energy consumption without VM migrations
g_n	The number of VM instances c_n ($n = \{1, 2, \dots, N\}$) accommodates.
λ	The data transmission rate between APs
ε_n	The resource utilization of c_n ($n = \{1, 2, \dots, N\}$)
$Y(t)$	The data transmission delay time
$\Theta_{n,i}(t)$	The running time of c_n ($n = \{1, 2, \dots, N\}$)
r_m	The requested number of VM instances.
s_m	The VM instances in the same cloud for executing the same task t_m could be composed as a special VM

Definition 3 (the data transmission rate between Aps). When the tasks and VMs are migrated between p_w and $p_{w'}$, there is a delay limited by the bandwidth between p_w and $p_{w'}$. We call it data transmission rate, denoted as $\lambda_{w,w'}$.

2.2. Energy Consumption Analysis. The energy consumption of cloudlets contains several aspects, including the energy consumption of active VMs, the energy consumed by the idle VMs, and the baseline energy consumption of running cloudlets. These three parts of energy consumption are calculated as follows [11].

The energy consumption of active VMs at the time instant t for the n th cloudlet c_n in C is calculated by

$$VE_n^{active}(t) = \sum_{i=1}^{g_n} \alpha_{n,i} \cdot \phi_{n,i}, \quad (1)$$

where $\alpha_{n,i}$ is the energy consumption rate of $v_{n,i}$ in active mode and $\phi_{n,i}$ is the total operating time of $v_{n,i}$.

In fact, the idle VMs with no task executions also have a baseline energy consumption. The idle time is determined by the running time of the longest running VM. The running time of the longest running VM is calculated by

$$\theta_{n,i}(t) = \max_{i=1}^{g_n} \{\phi_{n,i}\}, \quad (2)$$

Then the energy consumption of idle VMs in c_n is calculated by

$$VE_n^{idle}(t) = \sum_{i=1}^{g_n} \beta_{n,i} \cdot (\theta_{n,i}(t) - \phi_{n,i}), \quad (3)$$

where $\beta_{n,i}$ is the energy consumption rate of $v_{n,i}$ in idle mode.

When a cloudlet is active, there is at least one active VM running on it. It will consume the baseline energy. The energy for the all running cloudlets is calculated by

$$E_{cloudlet} = \sum_{n=1}^N \gamma_n \cdot \theta_{n,i}(t), \quad (4)$$

where γ_n is the energy consumption rate of c_n .

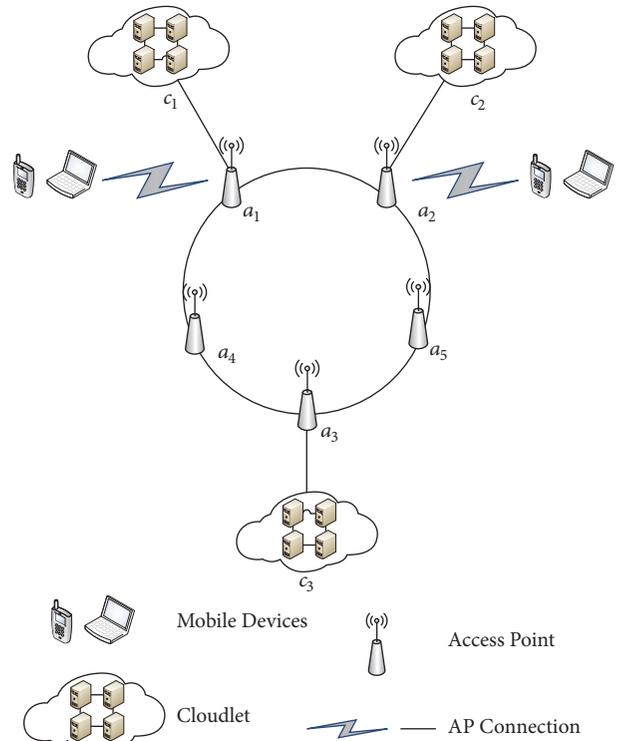


FIGURE 1: An example of WMAN with five APs and three cloudlets.

Through the above analysis, the total energy consumption is calculated by

$$E(t) = \sum_{n=1}^N VE_n^{active}(t) + \sum_{n=1}^N VE_n^{idle}(t) + E_{cloudlet}(t), \quad (5)$$

2.3. Resource Utilization Analysis. The resource utilization is a standard for measuring whether the resources are used sufficiently. When the resource utilization of c_n is low, the VMs in this cloudlet should be migrated to other cloudlets,

and then it could be shut down to save energy. The resource utilization of c_n which is denoted as ε_n is calculated by

$$\varepsilon_n = \frac{1}{g_n} \cdot \sum_{i=1}^{g_n} v_{n,i}(t), \quad (6)$$

where $v_{n,i}(t)$ is a binary variable employed to judge whether $v_{n,i}$ hosts a load, defined by

$$v_{n,i}(t) = \begin{cases} 0, & \text{if } \phi_{n,i} = 0, \\ 1, & \text{Otherwise.} \end{cases} \quad (7)$$

So, the average resource utilization of all the cloudlets is calculated by

$$\eta = \frac{1}{N'} \cdot \sum_{n=1}^N \varepsilon_n, \quad (8)$$

where N' is the number of all active cloudlets, calculated by

$$N' = \sum_{n=1}^N f_n(t), \quad (9)$$

where $f_n(t)$ is a binary variable employed to judge whether c_n accommodates a mobile task, defined by

$$f_n(t) = \begin{cases} 0, & \text{if } \sum_{i=1}^{g_n} \phi_{n,i} = 0, \\ 1, & \text{Otherwise.} \end{cases} \quad (10)$$

2.4. Delay and Privacy Preservation Analysis. Besides, when the data are transmitted between v_i and $v_{i'}$, there is always a delay due to the data transmission rate, which affects the experience of mobile users. The delay of all the tasks for data transmission among VMs is calculated by

$$Y(t) = \sum_{i=1}^{g_n} \sum_{i'=1}^{g_n} \frac{D_{i,i'}}{\lambda_{i,i'}}, \quad (11)$$

where $D_{i,i'}$ is the total size of data which are transferred from v_i to $v_{i'}$ and $\lambda_{i,i'}$ is the data transmission rate which is mentioned above.

When data are transmitted from one cloudlet to another, it is relayed through multiple APs. The APs in the WMAN have limited storage and computational capabilities, as well as low security, and the mobile users are vulnerable to violations while transmitting the applications and data with private personal information. An efficient and effective method to solve this problem is decreasing the number of APs that the transmitted data pass through. When there are fewer nodes which data passes through, the probability of privacy leakage is lowered; thus we could achieve the goal of privacy preservation.

3. A Cloudlet Management Method for Energy Conservation with Privacy Preservation

Based on the analysis of energy and the possibility of VM migrations for privacy preservation in WMAN in Section 2, an energy-efficient cloudlet management method is proposed in this section.

3.1. Method Overview. Technically, as specified in Box 1, our proposed cloudlet management method for energy saving and privacy preservation consists of five steps, i.e., the destination cloudlet identification, privacy-aware cloudlet confirmation, VM scheduling strategy obtaining, idle-state detection for cloudlets and global VM scheduling. The destination cloudlet identification is designed to confirm and record the active cloudlets with idle spaces, which could be employed to host the migrated VMs. These obtained cloudlets could be served as the input for the processing of Step 2. Step 2 is developed for the VMs in the source cloudlet to select the final cloudlet with the minimum data transmission time and, at the same time, achieve the purpose of privacy preservation. Step 3 is designed to produce the available migration strategies. Step 4 detects the idle state of active cloudlets in a dynamic way. All the achieved migration strategies and records are the input of the final global VM scheduling.

3.2. Destination Cloudlet Identification. When tasks are being hosted on a VM, each task may have a different requirement of idle VMs. The active cloudlets can host multiple VMs for task execution. For example, if a task requiring three virtual machines is migrated from another low-load virtual machine, the destination cloudlet must have more than or equal to three idle VMs. Otherwise, the task cannot be migrated to the currently selected cloudlet. Therefore, in Algorithm 1, we design the algorithm to find the migrated cloudlets which have the idle VMs and meet the requests. It plays an important role in cloudlet management.

Once the resource requirement of a task is responded by a VM instance, it generates a VM occupation record for the resource manager. Here, each cloudlet hosts only one PM. Let $OR = \{or_1, or_2, \dots, or_Z\}$ be the VM occupation record collection, where Z is the number of the records in OR , defined as follows.

Definition 4 (VM occupation record or_z). or_z records the occupation status of the task deployed in the cloudlet, which is denoted as $or_z = (tid_z, cid_z, pid_z, vid_z, stim_z, dtim_z)$, where tid_z , cid_z , pid_z , vid_z , $stim_z$, and $dtim_z$ are the allocated task, the occupied cloudlet, the relevant employed PM in the cloudlet, the applied VM instance, the service start time, and the duration time, respectively.

Here we select the VMs that meet the task requirements. To better schedule the VMs in an energy-efficient way, they are classified according to VM occupation records. These records generate from the destination cloudlets once current round of migration is finished.

Firstly, we get the running cloudlet list from or_z and then sort the cloudlets in the decreasing order of idle VMs; that is, we need to sort the different cloudlets according to the load distribution of the cloudlets. Suppose we want to select d cloudlets that meet the requirements. The cloudlets with higher load, meeting the resource requirements, are selected. Once the cloudlet meets the requirement, we add it to the selected running cloudlet list, denoted as $RS = \{rs_1, rs_2, \dots, rs_d\}$. When we traverse all the cloudlets in the

- Step1:** Destination cloudlet identification. Select a certain range of cloudlets with free VMs, which form a primitive cloudlet collection. This collection is identified for VM migration.
- Step2:** Privacy-aware cloudlet confirmation. Confirm the cloudlet with the shortest data transmission time for the source cloudlet from the cloudlet collection obtained in Step 1.
- Step3:** VM scheduling strategy obtaining. Migrating the tasks to the cloudlet obtained from Step 1 and Step 2. Update the migration policies and the task storage records.
- Step4:** Idle-state detection for cloudlets. Detect the idle state of each cloudlet, and calculate the resource utilization for the cloudlets, which is used as the migrated standard of dynamic scheduling.
- Step 5:** Global VM scheduling. Based on the VM migration strategies obtained in Step 3, the dynamic VM scheduling strategies for all the running VMs are achieved according to the cloudlet utilization obtained in Step 4.

Box 1: Specification of VM scheduling method for energy conservation.

Input: The acquired number q of idle VMs
VM occupation record or_z

Output: The selected running cloudlets $RS = \{rs_1, rs_2, \dots, rs_d\}$

- (1) Get the running cloudlets lists from or_z
- (2) Sort the PM in cloudlets in the decreasing order of idle VMs, $PM_s = \{pm_1, pm_2, \dots, pm_N\}$
- (3) $flag = 1, H = 0, n = 1$
- (4) **while** $flag = 1$ **do**
- (5) Get the number of idle VMs, i.e., rn
- (6) **if** r_n can hosts the VMs && $H < d$ **then**
- (7) Add r_n to rs_d
- (8) $n = n + 1$
- (9) $H = H + 1$
- (10) **else if** $n == N \parallel H == d$ **then**
- (11) $flag = 0$
- (12) **end if**
- (13) **end if**
- (14) **end while**
- (15) **Return** RS

ALGORITHM 1: Destination cloudlet identification DCI.

original list or find the number of qualified VMs that meet expectations, the searching processing would be stopped.

Algorithm 1 specifies the process of destination cloudlet identification. In this algorithm, we first sort the PMs in cloudlets in descending order of load (Lines (1) and (2)). Then we identify the cloudlets that satisfy the meets of idle VMs (Lines (3) to (14)).

3.3. Privacy-Aware Cloudlet Identification. Algorithm 1 identifies a range of cloudlets that meet the criteria. In this section, we use Dijkstra's algorithm to sort the selected cloudlets and then select the closest one to migration tasks from source cloudlet to the destination, with shortest path [20].

Besides, the processes of VM scheduling could lead to potential risks due to the privacy leakage in WMAN. In order to cater for the application scenario of WMAN, our algorithm is in demand to take privacy protection into consideration. In view of this challenge, we find that using the algorithm of shortest time identification based on Dijkstra is an appropriate method to achieve the goal of protecting users' privacy. When identifying the shortest data transmission time between source cloudlet and destination cloudlet, the data also go through the fewest APs; in other words, due to the

principle of Dijkstra, we reduce the possibility of privacy leaks.

Definition 5 (data transmission time dis_{ij}). The transmission time between the cloudlets i and j is denoted as dis_{ij} ($1 \leq i < j \leq d$) according to the transmission delay among the APs of these two cloudlets.

Firstly, all the vertices should be divided into two parts: i.e., the known shortest time vertex set P_d and the unknown shortest time vertex set Q_d . Initially, only one collection of the source cloudlet is known in the collection P_d of the shortest time. Secondly, the source cloudlet C_s is set by its own shortest time as 0, that is, $dis_{C_s} = 0$, if there is a vertex i that the source point can directly reach, set dis_i to y_{si} . The cloudlet with the smallest data transmission time among the collection Q_d is selected, and it should be added to the collection P_d . And all the edges starting from this cloudlet are examined. For example, if there is an edge from C_i to C_j , then the transmission time from C_s to C_j can be extended by adding the edge $C_i \rightarrow C_j$ to the tail. The length of this time is $dis_i + y_{ij}$. If this value is smaller than the currently known value of dis_j , we can replace the current value in dis_j with the new one.

Input: The selected running cloudlets $RS = \{rs_1, rs_2, \dots, rs_d\}$
 The known shortest time vertex set $P = \{p_1, p_2, \dots, p_d\}$
 The unknown shortest time vertex set $Q = \{q_1, q_2, \dots, q_d\}$
 The collections y_{ij} and dis_i
 The VM occupation record or_z

Output: The cloudlet with the shortest time C_{dij}

- (1) Get the transmission time between each cloudlet in RS from or_z to y_{ij}
- (2) Add the source cloudlet (C_s) to p and set its $dis = 0$
- (3) **if** C_i is directly connected to the source cloudlet **then**
- (4) $dis_i = y_{si}$
- (5) **end if**
- (6) sort Q in the increasing order of the transmission time
- (7) **for** each cloudlet in Q **do**
- (8) **if** $dis_i + y_{ij} < d_i$ **then**
- (9) $dis_i = dis_i + y_{ij}$
- (10) **end if**
- (11) **end for**
- (11) sort dis_i in the increasing order of the transmission time
- (12) $C_{dij} = d_i$
- (13) **return** C_{dij}

ALGORITHM 2: Privacy-aware cloudlet identification.

If the collection Q_d is empty, the algorithm ends. Then return the cloudlet with the shortest transmission time.

Algorithm 2 specifies the process of privacy-aware cloudlet identification. In this algorithm, we firstly divide all the vertices into two parts (Lines (1) and (2)); then we use Dijkstra algorithm to find the destination with the shortest time (Lines (3) to (11)). At last, we return the result (Lines (12) and (13)).

3.4. VM Scheduling Strategy Confirmation. In order to obtain the optimal energy savings, we try to migrate the tasks on the low-load cloudlets to the high-load cloudlets as much as possible, based on the results achieved by the above analysis. Once all the tasks on a cloudlet are migrated, the cloudlet could be changed to the sleep mode to save power. At the same time, a fully loaded cloudlet can also achieve higher resource utilization. And each migration procedure is recorded in the relevant VM occupation records with the migration time instant t .

Definition 6 (migration composition collection MC_d). For all the running VMs in rs_d , the synchronous VM migration strategies constitute the migration composition.

Suppose there are Z cloudlets in OR_Z and K migration compositions, i.e., $\{l_1, l_2, \dots, l_K\}$, for each turn of migration composition generation. First, we sort the cloudlets in the decreasing order of idle VMs, which means sorting from low load to high load. Here, flag is a sign. Get cloudlets to migrate from low load to high load from the list of cloudlets.

To facilitate the energy-efficient VM scheduling, the running VM list and hosting tasks in the occupation record set OR_z are obtained first. Then for each task in the list, we use Algorithms 1 and 2 to identify the most suitable

cloudlet for migration and migrate the task from the high-load cloudlet to the low-load cloudlet. And once a task in a low-load VM cannot be all migrated, the temporal migration strategies should be removed. For example, if a cloudlet hosts two tasks, the first task takes up one VM instance and the other takes up two VM instances. The two VMs instances of the second task should be placed in the same target cloudlet. Therefore, we should migrate the second task first. If the second task can be migrated but the first one does not work, leave the two tasks on the original cloudlet. Otherwise, the migration is successful and the temporary migration records $c_1 \sim c_k$ should be updated to the migration record collection MC_d , respectively.

Here, after the VM migrations at the time instant t , the start time and duration of the relevant VM occupation records should be updated, according to the real occupation time. Specifically, if the migration occurs, the duration of the task on the original cloudlet is necessary to be modified. For the task that is migrated, the relevant new VM occupation records are generated, the start time is updated to t , and the duration also needs to be changed as the predicted occupation time.

Algorithm 3 specifies the process of VM scheduling strategy confirmation. We firstly sort the cloudlets in the decreasing order according to the load (Line (1)). Then we move the tasks on the low-load cloudlets to the high-load cloudlets and store the records in the list migration record MC_d (Lines (2) to (21)).

3.5. Idle-State Detection for Cloudlets. The idle-state detection of active cloudlets is also one of the important factors for global migration considerations. It is a significant standard for the global VM scheduling. When the next migration is going to be conducted, we should evaluate the idle-state of

```

Input: The running cloudlet set  $rs_d$ ; VM occupation record  $or_z$ 
          VM occupation record  $or_z$ 
Output: The migration composition collection  $MC_d$ 
(1) Sort the cloudlets in the decreasing order of idle VMs
(2)  $flag = 0, z = 1, K = 1$ 
(3) while  $flag = 0$  do
(4)   Get the running VM list in  $ORz$ 
(5)   Get the hosting task list in  $ORz$ 
(6)   for each hosted task do
(7)     Get the number of occupied VMs
(8)     Select the destination cloudlet by Algorithms 1 and 2
(9)   end for
(10)  if all the VMs can be migrated away then
(11)    Generate a migration composition  $l_k$ 
(12)    Update the cloudlet status and VM distribution
(13)  else  $flag=1$ 
(14)    Delete the temporal migration strategies
(15)  end if
(16)     $z = z + 1, K = K + 1$ 
(17) end while
(18) for the reserved tasks' records do
(19)   $dtim_k = t$ 
(20) end for
(21) for  $k = 1$  to  $K$  do
(22)  Add  $l_1 \sim l_k$  as a migration composition to  $MC_d$ 
(23)  Get the start time and time of each migration task
(24)   $dtim_k = dtim_k - t$ 
(25)   $stim_k = t$ 
(26)  Update the start time and duration time in  $MC_d$ 
(27)  Update task storage record  $cs_x$ 
(28) end for
(29) Return  $MC_d$ 

```

ALGORITHM 3: VM scheduling strategy confirmation.

the active cloudlets, to confirm and record the idle space before the migration time t . After detecting the idle space, the current cloudlet resource utilization before the migration time could be achieved.

As the running time changes dynamically, some tasks would be competed over time. In this section, we detect the idle state of cloudlets as time progress. Therefore, to find the available destination cloudlet, the idle space and resource utilization of the all the running cloudlets should be detected first.

Once the migration strategy is designed in Algorithm 3, it generates one or more VM occupation records based on the migration records for the resource manager. So, we should also update the task storage records before the migration based on the latest migration strategy. Suppose there are x tasks that still need to be hosted. Let $CS = \{cs_1, cs_2, \dots, cs_x\}$ be the task occupation record collection, where x is the number of the records in CS , defined as follows.

Definition 7 (task storage record cs_x). cs_x records the storage status of the tasks deployed in the cloudlet, whose number is x , start time is denoted as $stim_x$, and duration time is denoted as $dtim_x$, which is dynamically changed.

The running cloudlet list from the task storage record cs_x is gotten first as the main data. Secondly, all the active tasks are traversed and the cloudlet idle space of all the cloudlets needs to be updated accordingly. Then, use the formula (9) to calculate the number of idle VM instances and the amount of the active cloudlets. At last, RP is returned as the final results of the idle spaces of the definite cloudlet.

Algorithm 4 specifies the process of idle-state detection of cloudlets. Firstly, we update the cloudlet idle space (Lines (1) to (6)). Then we calculate the resource utilization and return it (Lines (7) and (8)).

3.6. Global VM Scheduling. To achieve the goal of energy-efficient cloudlet management in WMAN, the optimal VM migration composition should be confirmed from the migration strategies which is achieved by Algorithm 3. In this section, global VM scheduling is designed to determine which VMs and where should be migrated for the whole execution period of the VMs.

Due to the goal of scheduling all the running VMs from a global perspective in this section, migration operation time is necessary for monitoring the resource utilization of all the cloudlets through the calculating of the current VM

Input: The task storage record cs_x
Migration time t

Output: The current cloudlet resource utilization RP

- (1) Get running cloudlet list from the cs_x of time t
- (2) **for** each hosted task **do**
- (3) Get the number of occupied VMs
- (4) **end for**
- (5) Update the cloudlet idle space
- (6) **end for**
- (7) Calculate the current cloudlet resource utilization RP by (9)
- (8) Return RP

ALGORITHM 4: Cloudlet idle-state detection.

Input: The VMs $V = \{v1, v2, \dots, vz\}$
Migration time t
Total operation time H

Output: The cloudlets scheduling policy

- (1) Get the migration time t ;
- (2) Get the number of idle VMs on each cloudlets
- (3) **while** $t < H$ **do**
- (4) Get RP from Algorithm 4
- (5) **if** $RP < \text{utilization threshold level}$ **then**
- (6) Update the migration time t as the next finish time of the VMs.
- (7) Get migration compositions MC_h by Algorithm 3
- (8) Schedule the VMs by the policy
- (9) Set the vacant cloudlets to the sleeping mode
- (10) **end if**
- (11) **end while**

ALGORITHM 5: Global VM scheduling.

occupation records. The number of idle VMs of each cloudlet can be detected dynamically. Algorithm 4 is designed to acquire the idle space as the migration standard which is one of the inputs in Algorithm 5.

As we aim to schedule the VMs with the goals of energy saving and privacy preservation, the VM migration composition with highest utilization will be chosen as the ultimate VM scheduling policy. And after VM scheduling for the identified VMs, the vacant PMs should be set to the sleeping mode to save the energy. Besides, through the Algorithm 2, we can also attain another purpose for reducing the possibility of privacy leaks.

The migration times are necessary to be obtained for global VM scheduling according to the utilization threshold. Besides, the number of the idle VMs on each cloudlet is a key measure to decide the destination cloudlet of the migrations. During the execution period of all the mobile applications in WMAN, the resource utilization RP of the cloudlets is obtained by Algorithm 4. Then, if the current RP is less than utilization threshold, that is, there are enough space in the active cloudlets after the previous round of scheduling, we need to schedule the VMs by the policy and set the vacant cloudlets to the sleeping mode according to MC_h by Algorithm 3. Thus, we achieve the goal of dynamic scheduling.

Algorithm 5 specifies the process of global VM scheduling (t). Firstly, we update the RP and MC_h of the migration time (Lines (1) to (6)). Then schedule the running VMs and set the vacant cloudlets to the sleeping mode (Lines (8) to (10)).

4. Experiment Evaluation

In this section, a series of comprehensive experiments are operated to evaluate the performance of our proposed energy-efficient cloudlet management method for privacy preservation in WMAN. For comparison analysis, the running state without migration is marked as a benchmark is employed to verify the effectiveness of our proposed ECM method.

4.1. Experimental Context. In our experiment, we select the specified type of cloudlet servers to create our simulated cloud infrastructure services net and the basic hardware parameter are as follows. Its configuration consists of Intel Xeon 3040, dual-Processor clocked at 1860 MHz, and 4 GB of RAM. Its baseline power consists of two parts, the PM part and the VM part. The power consumption of the PM, active VM, and idle VM are 86W, 6W, and 4W, respectively.

Six basic parameters are initialized as records in our experiment. The value domain of each parameter is specified

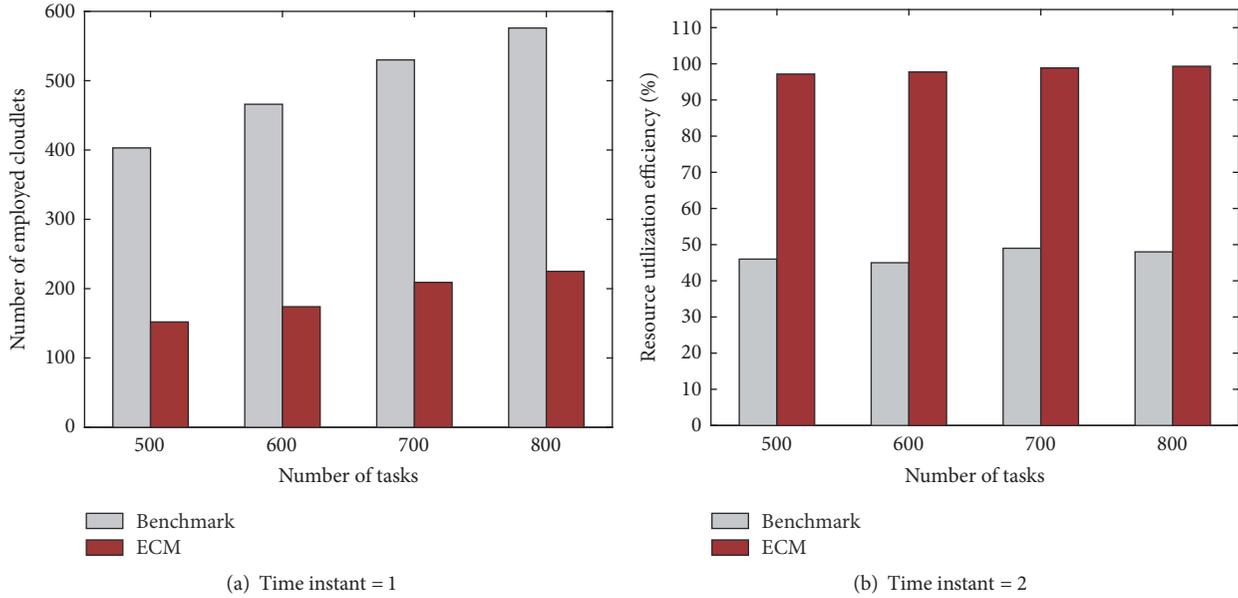


FIGURE 2: Comparison of the number of employed cloudlets for different time instants with Benchmark and ECM.

TABLE 2: Parameter settings.

Parameter Item	Domain
Number of running PMs	{500, 600, 700, 800}
Number of running VMs in each PM	[1, 3]
Number of VMs on each PM	6
Data transmission rate (Mb/s)	[270, 540]
VM duration time	{1, 3}
VM transmission data (G)	[0.5, 0.8]

in Table 2. In other words, 4 different-scale datasets are generated, respectively, for our experiment with the number of running tasks 500, 600, 700, and 800. Each data record in these datasets is a tuple with 6 attributes. For example, there is a data record (6, 18, 2, 0, 3, and 2), where “6” is the sequence number of the hosted task, “18” is the cloudlet sequence number, “2” is the number of VMs that this hosted task simultaneously occupies, “0” is the VM start time, “3” is the VM duration time, and “2” is the total amount of data for this hosted task.

In our experiment, there are two rounds of multiple VM migration, and we use time instants before migration to distinguish different round of VM scheduling. The two time instants are recorded as 1 and 2 in the experimental section, respectively.

4.2. Performance Evaluation. In this section, evaluations on energy consumption, resource utilization, and transmission delay are discussed to validate our proposed method.

4.2.1. Performance Evaluation on Energy Consumption. Our energy consumption evaluation is based on the energy consumption of the running cloudlets and VMs. The number of active employed cloudlets is also significant to the total

energy consumption. We used four different sets of data to compare energy consumption in the case of benchmark and ECM. And we use two time instants to compare the employed number of different time instants in the process of dynamic scheduling. From Figure 2, it can be concluded that, with the ECM method, the number of running cloudlets is much less than that in the benchmark case in both two time instants. Since the idle cloudlet can be directly considered to be completely closed, its energy consumption can be ignored in our experiment when setting them to the sleeping mode. Thus, less active cloudlet can generate less energy. The energy consumption here is divided into three parts, the idle VMs, the active VMs, and the energy consumed by active cloudlets themselves.

After two rounds of VM migrations, the total energy consumption can be calculated respectively. Figure 3 indicates these energy values in the case of benchmark and ECM. Table 3 shows the results of improvements of energy consumption with ECM compared to benchmark. From the analysis of Figure 3 and Table 3, using our proposed method ECM saves more energy than using benchmark.

4.2.2. Performance Evaluation on Resource Utilization. Resource utilization efficiency for two different time instants is also one of the dimensions that measure the value of our proposed method. Resource utilization changes dynamically in different rounds of VM migrations because when the resource utilization of one cloudlet is low, the VMs in this cloudlet should be migrated to other cloudlets, and then it could be shut down to save energy. It refers to the relationship between VMs and active cloudlets. Similar to the energy consumption evaluated above, resource utilization is also tested by four different datasets. The ECM and benchmark are compared with these four datasets, and the simulated resource utilization result is illustrated in Figure 4.

TABLE 3: Improvements of energy consumption with ECM compared to benchmark.

	Number of servers			
ECM	500	600	700	800
	54.4%	54.1%	41.9%	52.1%

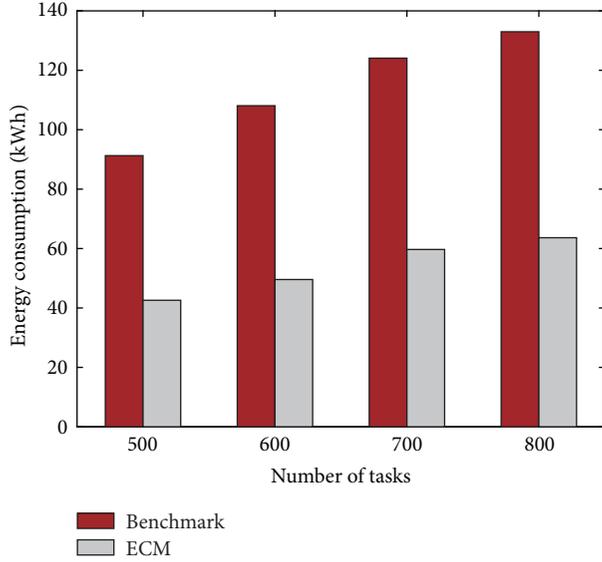


FIGURE 3: Comparison of total energy consumption with Benchmark, ECM.

As can be shown in the chart, after VM migrations, our proposed method ECM has better resource utilization than benchmark. And, with the increasing times of VM scheduling, the resource utilization for cloudlets is getting higher. Since ECM considers the energy consumption by cloudlets, some VMs would not be migrated to other cloudlets and the cloudlet hosting it still keeps in active mode, which would raise the resource utilization. It shows that, with our method for energy saving, the resource utilization becomes higher than the benchmark.

4.2.3. Performance Evaluation on Transmission Delay. Figure 5 illustrates the observation results of the amount of the migrations after VM scheduling. According to the system model, the transmission delay is determined based on the amount of data and the data transmission rate. At the same time, the number of VM migrations is also one of the important factors of total transmission delay. Compared to benchmark, our proposed method ECM uses a large amount times of VM migrations to achieve the goal of saving energy which would generate high transmission delay.

Furthermore, to make the evaluation on performance of our proposed ECM method more clearly, the effect of VM migrations on transmission delay is fully investigated in detail. The transmission delay of our proposed ECM method is based on four different data sets and two time instants for simulation evaluation, respectively.

The result which illustrates transmission delay time caused by the VM migrations is evaluated. From Figure 6,

simulation results can be intuitively achieved that the transmission delay based on the ECM method fluctuates around 10 minutes in the simulation environment. The result shows that there is still a lot of room for improvement in handling challenges of high transmission delay.

Besides, from the results of evaluations on VM migrations, we can achieve that using the shortest time identification can indeed reduce the number of migrations; in other words, we can also achieve the goal of reduces the possibility of privacy leaks.

5. Related Work

Recently, the emergence and development of WMAN aim to meet the growing market requirements of wireless broadband. An increasing number of mobile users in urban cities have tendency to accomplish the execution of various tasks in clouds due to the abundant resources, but one significant limitation of offloading tasks to the remote clouds is the long physical distance between mobile users and remote cloud [21–23]. To reduce such delay of offloaded tasks to remote clouds, cloudlets are deployed in WMAN to support mobile devices, by executing offloaded tasks on local cloudlets.

Ma et al. [24] realized the placement of cloudlets in WMAN is important and proposed a New Heuristic Algorithm (NHA) to reduce the sorting process of APs running time to solve the delay problem. Xiang et al. [25] proposed an adaptive cloud placement method for GPS big data mobile applications that adjusts the placement of the cloud by identifying the aggregated area of mobile devices. Mike Jia et al. [26] proposed an algorithm of allocating the requirements of mobile users to suitable cloudlets deployed in WMAN to implement load balancing to solve the delay problem and discussed how their algorithm can be practically applied to WMANs with dynamic and constantly moving users.

Currently, live VM migration techniques significantly make contribution to manage resource in cloudlet, thus improving energy efficiency and effectively lower the delay in cloudlets as much as possible via providing high quality of resources for execution of tasks and migrating VMs to idle cloudlets to reduce task queuing time.

Decreasing the delay in cloudlets could be divided into two aspects: decreasing transmission delay between cloudlets and decreasing processing delay in cloudlets [27]. But Rodrigues et al. [28] reduced the two kinds of delay at the same time by utilizing a mathematical model with a Particle Swarm Optimization (PSO) model. Dhanoa et al. [29] quantified the energy consumption and time in the VM migration process and mathematically modeled them and then discovered that the main factors of energy consumption during VM migration are the size of the VMs and the network bandwidth. Xu et al. [30] proposed a VM scheduling

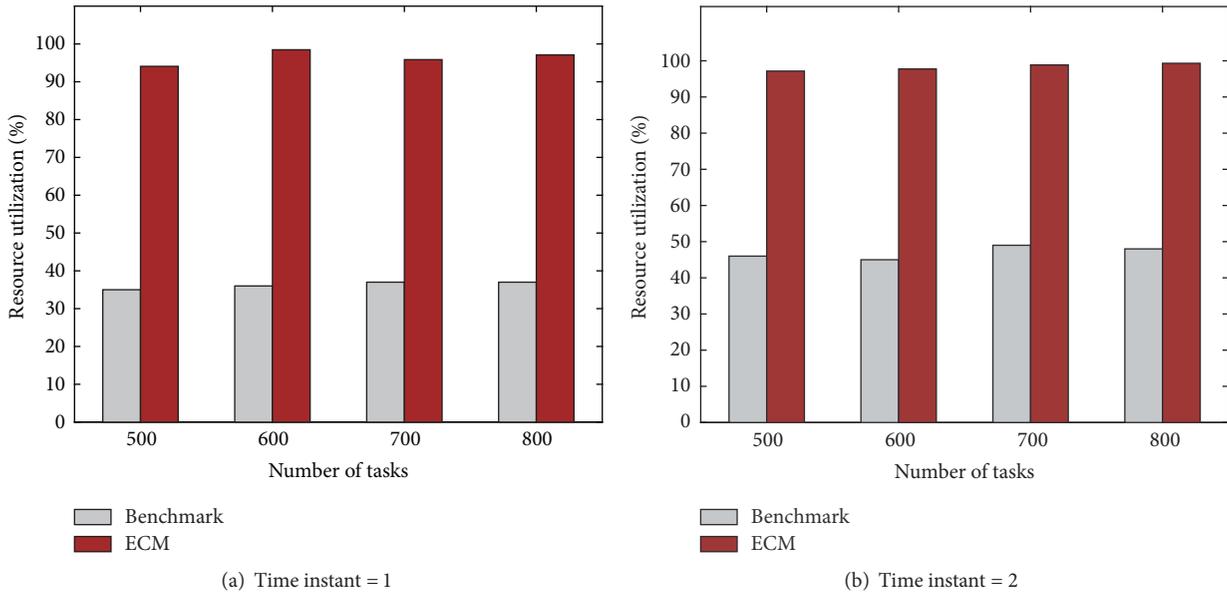


FIGURE 4: Comparison of resource utilization for different time instants with Benchmark, ECM.

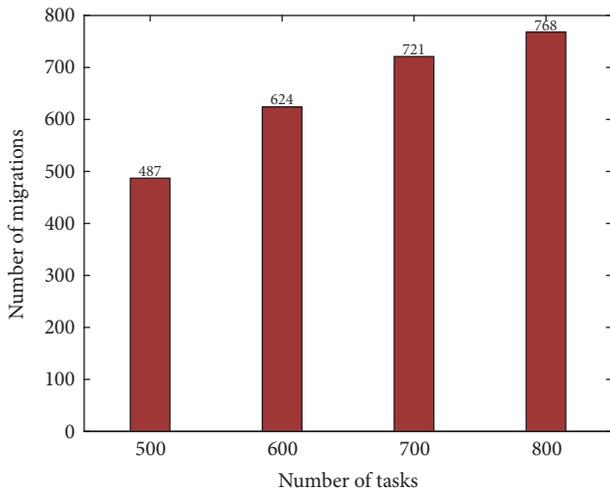


FIGURE 5: Number of migrations with ECM.

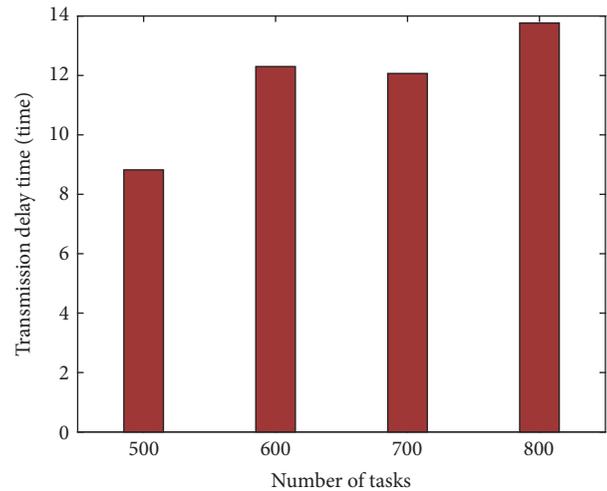


FIGURE 6: Transmission delay time of ECM.

method to achieve balance between energy and performance in cyberphysical cloud systems and the method determines which VMs and where should be migrated. Shaw et al. [31] proposed a method to save energy by using a VM migration dynamic merge to reduce the number of running physical machines and the authors designed an algorithm to determine if the VM really needs to continue running or if it needs to be migrated.

The optimization problem of energy consumption for cloudlet management could be solved by the heuristic methods and the convex optimization technique [32, 33]. Besides, along with the other scheduling objectives, the game theory with Nash Equilibria is employed to realize the competition usage of the cloudlet resources [34, 35]. However, some cloudlet services need to access mobile users' personal private

information. When transmitting the mobile applications of the mobile users, personal privacy is easily leaked through some APs with poor security, which may cause some bad events and even hurt the personal safety of mobile users. Therefore, privacy preservation could be treated as a constraint for the optimization problem in cloudlet management [36–39].

Maria et al. [40] proposed a conceptual framework in Ambient Intelligent focused in cloud computing that includes the various privacy policies and privacy issues. Ji et al. [41] implemented a hybrid privacy protection solution with KP-ABE and CP-ABE, whose advantage is balancing the performance and security in cloud platforms. Wang et al. [42] developed a method to protect privacy by dividing user's private information into several parts and then this

system saves the data in different cloud servers and user's fog devices. Moreover, when the data itself is insecure, which may include trust spoof attacks and replicated-sink attacks, the data collection needs to identify whether such data could be received. Wang et al. [43] proposed a trust evaluation scheme to ensure the trustworthiness of sensors and mobile sinks, which protects the system from trust spoof attacks and replicated-sink attacks.

The above analysis concludes that live VMs migration technique has not been well used in the cloudlets researches and few studies have considered privacy preservation as much as possible while reducing energy consumption about cloudlets deployed in WMAN.

6. Conclusion and Future Work

The utilization of cloudlet in WMAN is an important technology which extends the performance of mobile devices. As WMAN continues to develop, cloudlets which are public and easy to access would be vital to mobile cloud computing.

In this paper, we proposed an energy-efficient cloudlet management for privacy preservation in WMAN abbreviated as ECM taking advantage of the live VM migration techniques. We implemented an effective algorithm that determines the place to which VMs should be migrated. Our goal is to reduce the energy consumption of cloudlets and privacy preservation in WMAN. Experimental evaluations have been conducted to validate the efficiency and effectiveness of our proposed method. As this paper mainly focused on the energy saving and privacy in WMAN, there are also many important issues which need to be investigated in future. For example, in the people-intensive areas, there are large amounts of mobile applications that need to be offloaded to the cloudlets for execution; how to reduce the energy consumption and achieve load balancing of the cloudlets in WMAN remains challenging. When we only pay attention to energy saving, the communication delay would be high, which affects the experience of mobile users.

For future work, we try to take cloudlet load balancing and energy consumption of cloudlets into consideration at the same time. Furthermore, we will design a corresponding method for trade-offs between energy consumption and load balancing.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the National Science Foundation of China (under Grant nos. 61702277, 61672276, 61501247, 61772283, 61402167, and 61672290), the Key Research and Development Project of Jiangsu Province (under Grant nos.

BE2015154 and BE2016120), and the Natural Science Foundation of Jiangsu Province (under Grant no. BK20171458). Besides, this work is also supported by The Startup Foundation for Introducing Talent of NUIST, the open project from State Key Laboratory for Novel Software Technology, Nanjing University under Grant no. KFKT2017B04, the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAET), and the project "Six Talent Peaks Project in Jiangsu Province" under Grant no. XYDXXJS-040.

References

- [1] S. B. M. Baskaran and G. Raja, "Blind key distribution mechanism to secure wireless metropolitan area network," *CSI Transactions on ICT*, vol. 4, no. 2-4, pp. 157–163, 2016.
- [2] H. M. T. Al-Hilfi, A. H. Najim, and A. M. Alsahlany, "Evaluation of WIMAX network performance of Baghdad city using different audio codecs," in *Proceedings of the 16th RoEduNet Conference: Networking in Education and Research, RoEduNet '17*, pp. 1–5, 2017.
- [3] J. Cheng, R. Xu, X. Tang et al., "An abnormal network flow feature sequence prediction approach for DDoS attacks detection in big data environment," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 95–119, 2018.
- [4] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.
- [5] V. Meena, M. HariPrasath, V. Kalpana, K. ArunKumar, and J. SenthilKumar, "Optimal resource reservation for offloaded tasks in mobile cloud computing," in *Proceedings of the 2nd International Conference on Communication and Electronics Systems (ICCES)*, pp. 677–682, 2017.
- [6] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM '16*, pp. 1–9, 2016.
- [7] L. Li, F. Meng, and P. Ju, "Some new integral inequalities and their applications in studying the stability of nonlinear integro-differential equations with time delay," *Journal of Mathematical Analysis and Applications*, vol. 377, no. 2, pp. 853–862, 2011.
- [8] F. Teka, C.-H. Lung, and S. Ajila, "Seamless Live Virtual Machine Migration with Cloudlets and Multipath TCP," in *Proceedings of the 39th IEEE Annual Computer Software and Applications Conference, COMPSAC '15*, vol. 2, pp. 607–616, 2015.
- [9] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Capacitated cloudlet placements in wireless metropolitan area networks," in *Proceedings of the 40th Conference on Local Computer Networks, LCN '15*, pp. 570–578, 2015.
- [10] S. Guan, R. E. De Grande, and A. Boukerche, "A Cloudlet-based task-centric offloading to enable energy-efficient mobile applications," in *Proceedings of the IEEE Symposium on Computers and Communications, ISCC '17*, pp. 564–569, 2017.
- [11] X. Xu, W. Dou, X. Zhang, and J. Chen, "An energy-aware resource allocation method for scientific workflow executions in cloud environment," *IEEE Transactions on Cloud Computing*, vol. 4, no. 2, pp. 166–179, 2016.
- [12] M. S. Shinde and K. M. Ashtankar, "Effect of different shapes of conformal cooling channel on the parameters of injection

- molding computers,” *Computers, Materials & Continua*, vol. 54, no. 3, pp. 287–306, 2018.
- [13] T. Wang, M. Z. A. Bhuiyan, G. Wang, M. A. Rahman, J. Wu, and J. Cao, “Big data reduction for a smart city’s critical infrastructural health monitoring,” *IEEE Communications Magazine*, vol. 56, no. 3, pp. 128–133, 2018.
- [14] C. Wu, E. Zapevalova, Y. Chen, and F. Li, “Time optimization of multiple knowledge transfers in the big data environment,” *Computers, Materials & Continua*, vol. 54, no. 3, pp. 269–285, 2018.
- [15] Z. Jia, X. Wei, H. Guo, W. Peng, and C. Song, “A privacy protection strategy for source location in WSN based on angle and dynamical adjustment of node emission radius,” *Journal of Electronics*, vol. 26, no. 5, pp. 1064–1072, 2017.
- [16] C. Jinhua, Z. Yuanyuan, C. Zhiping, L. Anfeng, and L. Yangyang, “Securing display path for security-sensitive applications on mobile devices,” *Materials & Continua*, vol. 55, no. 1, pp. 17–35, 2018.
- [17] L. Yuling, H. Peng, and J. Wang, “Verifiable Diversity Ranking Search Over Encrypted Outsourced Data,” *Computers, Materials & Continua*, vol. 55, no. 1, pp. 037–057, 2018.
- [18] L. Donghui, Z. Guozheng, X. Zeng, L. Yong et al., “Modelling the Roles of Cewebrity Trust and Platform Trust in Consumers’ Propensity of Live-Streaming: An Extended TAM Method,” *Computers, Materials & Continua*, vol. 55, no. 1, pp. 137–150, 2018.
- [19] L. Wenyan, L. Xiangyang, L. Yimin, L. Jianqiang, L. Minghao, and Q. S. Yun, “Localization Algorithm of Indoor Wi-Fi Access Points Based on Signal Strength Relative Relationship and Region Division,” *Computers, Materials & Continua*, vol. 55, no. 1, pp. 71–93, 2018.
- [20] J.-Q. Cai and H. Li, “An implicit degree condition for relative length of long paths and cycles in graphs,” *Acta Mathematicae Applicatae Sinica*, vol. 32, no. 2, pp. 365–372, 2016.
- [21] Y. Cao, Z. Zhou, X. Sun, and C. Gao, “Coverless information hiding based on the molecular structure images of material,” *Computers, Materials and Continua*, vol. 54, no. 2, pp. 197–207, 2018.
- [22] C. Yuan, X. Li, Q. M. J. Wu, J. Li, and X. Sun, “Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis,” *Computers, Materials and Continua*, vol. 53, no. 4, pp. 357–371, 2017.
- [23] J. Kaur and K. Kaur, “A fuzzy approach for an IoT-based automated employee performance appraisal,” *Computers, Materials and Continua*, vol. 53, no. 1, pp. 24–38, 2017.
- [24] L. Ma, J. Wu, and L. Chen, “Fast algorithms for capacitated cloudlet placements,” in *Proceedings of the 21st IEEE International Conference on Computer Supported Cooperative Work in Design, CSCWD ’17*, pp. 439–444, 2017.
- [25] H. Xiang, X. Xu, H. Zheng et al., “An adaptive cloudlet placement method for mobile applications over GPS big data,” in *Proceedings of the 59th IEEE Global Communications Conference, GLOBECOM ’16*, pp. 1–6, 2016.
- [26] M. Jia, J. Cao, and W. Liang, “Optimal Cloudlet Placement and User to Cloudlet Allocation in Wireless Metropolitan Area Networks,” *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, 2017.
- [27] T. Wang, J. Zeng, Y. Lai et al., “Data collection from WSNs to the cloud based on mobile Fog elements,” *Future Generation Computer Systems*, 2017.
- [28] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, “A PSO model with VM migration and transmission power control for low Service Delay in the multiple cloudlets ECC scenario,” in *Proceedings of the 2017 IEEE International Conference on Communications, ICC ’17*, pp. 1–6, 2017.
- [29] I. S. Dhanoa and S. S. Khurmi, “Analyzing energy consumption during VM live migration,” in *Proceedings of the International Conference on Computing, Communication and Automation, ICCCA ’15*, pp. 584–588, 2015.
- [30] X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, and S. Yu, “A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems,” *Future Generation Computer Systems*, 2017.
- [31] S. B. Shaw and A. K. Singh, “Use of proactive and reactive hotspot detection technique to reduce the number of virtual machine migration and energy consumption in cloud data center,” *Computers and Electrical Engineering*, vol. 47, pp. 241–254, 2015.
- [32] P. Wang and L. Zhao, “Some geometrical properties of convex level sets of minimal graph on 2-dimensional Riemannian manifolds,” *Nonlinear Analysis: Theory, Methods & Applications*, vol. 130, pp. 1–17, 2016.
- [33] P. Li, S. Zhao, and R. Zhang, “A cluster analysis selection strategy for supersaturated designs,” *Computational Statistics & Data Analysis*, vol. 54, no. 6, pp. 1605–1612, 2010.
- [34] J. Zhang, B. Qu, and N. Xiu, “Some projection-like methods for the generalized Nash equilibria,” *Computational Optimization and Applications*, vol. 45, no. 1, pp. 89–109, 2010.
- [35] B. Liu, B. Qu, and N. Zheng, “A successive projection algorithm for solving the multiple-sets split feasibility problem,” *Numerical Functional Analysis and Optimization*, vol. 35, no. 11, pp. 1459–1466, 2014.
- [36] H. Wu, Y. Ren, and F. Hu, “Continuous dependence property of BSDE with constraints,” *Applied Mathematics Letters*, vol. 45, pp. 41–46, 2015.
- [37] S. Lian and Y. Duan, “Smoothing of the lower-order exact penalty function for inequality constrained optimization,” *Journal of Inequalities and Applications*, 2016.
- [38] Y. Wang, X. Sun, and F. Meng, “On the conditional and partial trade credit policy with capital constraints: A Stackelberg Model,” *Applied Mathematical Modelling*, vol. 40, no. 1, pp. 1–18, 2016.
- [39] H. Li and S. Wang, “Partial condition number for the equality constrained linear least squares problem,” *Calcolo. A Quarterly on Numerical Analysis and Theory of Computation*, vol. 54, no. 4, pp. 1121–1146, 2017.
- [40] M. Del Mar López Ruiz and J. Pedraza, “Privacy risks in cloud computing,” in *Intelligent Agents in Data-Intensive Computing*, vol. 14, pp. 163–192, Springer International Publishing, 2015.
- [41] J. Yi-mu, K. Jia-bang, L. Hai et al., “A privacy protection solution for the balance of performance and security in cloud computing,” in *Proceedings of the International Conference on Web Information Systems Engineering – WISE 2013 Workshops*, pp. 335–345, 2014.
- [42] T. Wang, J. Zhou, M. Huang et al., “Fog-based storage technology to fight with cyber threat,” *Future Generation Computer Systems*, vol. 83, pp. 208–218, 2018.
- [43] T. Wang, Y. Li, W. Fang et al., “A comprehensive trustworthy data collection approach in sensor-cloud system,” *IEEE Transactions on Big Data*, p. 1, 2018.

